Kompütasyonel
Düşünme
Biçimini
Destekler

STEAM
•Science •Technology •Engineering •Art •Mathematics

24. BASKI

MEB MÜFREDATINA UYGUNDUR

Tüketmekten Üretmeye
MAKER
Eğitim Hareketi
Sürdürülebilir Bir Model Önerisi

Çocuklar için SCRATCH ile kodlama

Hakan ATAŞ
Bager AKBAY

Çocuklar için Scratch ile Programlama

9+Yaş 1. Kitap

Hakan ATAŞ - Bager AKBAY

abaküs

abaküs 018
Çocuklar için Scratch ile Programlama 9+ Yaş 1. Kitap
Hakan ATAŞ - Bager AKBAY
1. Baskı: Eylül 2015
2. Baskı: Mayıs 2016
3.-10. Baskı: Ağustos 2016
11.-17. Baskı: Temmuz 2017
18.-20. Baskı: Eylül 2017
21.-22. Baskı: Temmuz 2018
23.-24. Baskı: Kasım 2018
ISBN: 978-605-9129-21-3

Maker Eğitim Kitaplığı Dizi Editörleri: Bager Akbay - Hakan Ataş

Editör:	Barış Yücel Turan
Kapak ve sayfa düzeni:	Cem Demirezen-Cansu Yalçıner
Düzelti:	Özcan Oğuz
Satış:	satis@abakuskitap.com
Baskı:	Deniz Ofset Matbaacılık Maltepe Mah. Hastane Yolu Sok. No:1/6 Zeytinburnu-İstanbul Tel: 0212 613 30 06 - Faks: 0212 613 51 97 Matbaa sertifika No: 40200

Bu kitabın bütün yayın hakları Abaküs Kitap Yayın Pazarlama'ya aittir. Yayınevimizin yazılı izni olmaksızın kısmen veya tamamen alıntı yapılamaz, kopya edilemez, çoğaltılamaz ve yayınlanamaz.
Kitapta kullanılan logolar, firmaların tescilli logolarıdır.

Selda Ustabaş Demiryakan
Abaküs Kitap Yayın Dağıtım Hizmetleri
Yayıncılık sertifika no: 31359
Hobyar Mah. Cemal Nadir Sok. No:24/227 Cağaloğlu/Fatih-İST.
www.abakuskitap.com - editor@abakuskitap.com
Tel.: (0212) 514 68 61

Hakan ATAŞ

Hakan Ataş'ın matematik ve programlamaya olan merakı çocukluk yıllarında başladı. Fen Lisesi yıllarında aldığı yoğun matematik ve programlama eğitimlerinde matematik ağır bastı ve üniversite sınavı sonucunda Boğaziçi Üniversitesi Matematik bölümüne yerleşti. 2000 yılında mezun oldu ve ardından 2 yıla yakın bir süre Bilgisayar Programcısı olarak çalıştı. Bu 2 yılın ardından, kariyerine eğitimci olarak devam etme kararı aldı. Çeşitli kurumlarda Matematik öğretmenliği yaptı. Bu yıllarda birçok ders kitabı ve deneme kitapçığı yazdı.

2010 yılından itibaren, çocuklarla programlama ve etkileşimli içeriklerin üretimi üzerine çalışmalarına başladı. Bu konuda çeşitli eğitim kurumlarında eğitimler düzenledi. 2014 yılında tanıştığı Maker Hareketi ile, çalışmalarına ivme kazandırdı. Şu an, özel bir okulda eğitimde teknoloji entegrasyonu çalışmalarına devam etmektedir.

Bager AKBAY

Bager Akbay, Atatürk Fen Lisesi'nde geçirdiği matematik ve programlama ile dolu üç yılın ardından kazandığı eğitim bursu sayesinde bir sene Tokyo'da okudu. İstanbul'a döndükten sonra İTÜ Matematik Mühendisliği'ne girdi. Linux ile tanıştığı okulunda, bir yandan güvenlik, ağ ve programlama üzerine çalışırken bir yandan da Kara Tiyatro adlı amatör bir tiyatro grubunda oyunculuk yapmaya ve kukla oynatmaya başladı. Tiyatro deneyiminin verdiği güç ile, okulu bırakıp, tasarım okumak için Yıldız Teknik Üniversitesi'ne geçti. Programlama ile tasarımın ilişkisini keşfettiği bu yıllarda sektöre atıldı ve birçok ajans için etkileşimli içerikler üretmeye başladı.

Mezun olur olmaz öğretim görevlisi olarak çalışmaya başladı. 4 yıllık eğitimcilik deneyiminin ardından, Avusturya'ya dijital sanat okumaya gitti. Fiziksel etkileşim, yapay zekâ ve sanat üzerine birçok çalışma yaptı. Kızının doğumu vesilesiyle Türkiye'ye döndü ve eğitimcilik kariyerine devam etme kararı aldı. 4 sene boyunca sürdürdüğü eğitimciliğin yanı sıra, sanat projeleri üretmeye ve danışmanlığa devam ediyor.

Yoğun bir dönemin ardından yerel biri olmaya karar verdi ve Kadıköy Yeldeğirmeni'nde, projelerini hayata geçirebilmek için arkadaşlarıyla beraber İskele47 adında bir atölye açtı.

ÖNSÖZ

2014 yılının sıcak bir Temmuz günü. Çocukluğumun o güvenli mahalle arkadaşlığı ortamının özellikle büyükşehirlerde izinin bile kalmadığı, çalışan anne-baba, trafik ve azalan kamusal alan sarmalında apartman dairelerine sıkışıp kalan 21. yüzyıl çocuklarından biri olan 12 yaşındaki kızımla evde oturmuş ve her zamanki gibi elimizdeki ekranlara gömülmüştük.

Her "proje çocuğu" gibi doğduğundan bu yana müzik, resim, spor, vb. kurslara yazılıp anne-babasının zoruyla bunlara bir süre devam eden kızımı bu yaz rahat bırakmaya karar vermiştim ki, bir dostumun çocuklara yönelik programlama dillerini paylaştığı tweetini görünce son bir kez şansımı denemeye karar verdim. Konuyu ona açmadan sosyal medya üzerinden "çocuklar için programlama" eğitimi olup olmadığını soran bir paylaşım yaptım ve dikkatli bir arkadaşımın yanıtı sayesinde (çok büyük bir tesadüf eseri), 1 hafta sonra hemen yakınımızda başlayacak ücretsiz bir yaz kursu yakaladım.

Kızıma konuyu çekinerek açtım ama tereddütsüz kabul etmesine de şaşırmadım değil! Nedenini ise daha ilk derslerini izlerken anladım, hem dijital yerli kuşağın kullanıcı olarak bu konuya doğal bir yatkınlığı var hem de kullanılan programlama kodları (en azından başlangıç seviyesindekiler) bizim zamanımızdaki gibi siyah ekran üzerine yazılmış binlerce beyaz satır yerine lego parçalarına benzer renkli görsel öğelerden oluşuyor ve bu sayede çocuklar için programlama bilgisayar ekranındaki bir oyundan farksız hale geliyor (ki bu da onların üretkenliğini muazzam biçimde artırıyor!).

Ancak bunun da ötesinde kodlamayı öğreten eğitmenler; çocukları toplayıp tahta ya da ekranda bir şeyler anlatmak yerine, onlara derse gelmeden önce özel ev ödevleri veriyor ve öğrenciler bu verilen ödevleri yapıp kursa geldiklerinde de hepsiyle ayrı ayrı ilgilenerek, bir koç misali her birine takıldıkları noktalarda yardımcı olarak onları çözüm noktasına doğru taşıyorlar. Yeni eğitim sistematiğinde "ters yüz edilmiş sınıf (flipped classroom)" adı verilen modelin hayata geçirilmiş hali bu.

Sonuç olarak kızım (diğer arkadaşları gibi), 2 ay gibi kısa bir sürede ilk mobil uygulamasını yazdı ve Google Play Store'a yükledi. Bu, hem onun özgüvenine ciddi bir katkı yaptı hem de orada tanıştığı çocuklarla son derece pozitif bir etkileşime girdi. Bunun da ötesinde, artık kendisini bu yeni dünyanın bir parçası olduğunu çok daha fazla hissediyor.

Kadıköy-İskele47 ekibi tarafından verilen bu eğitim modeli aslında İrlanda'da doğan ve yayılan **CoderDojo** adlı küresel ve sivil bir eğitim insiyatifi. Hedefi ise, (gelecekte hangi mesleği seçerlerse seçsinler) yeni kuşak bireyler için artık kaçınılmaz hale gelen programlamanın temelini atmak ve bu temeli alan öğrencilerin ilk aşamada programlayıcı (coder) ama daha sonra uygulayıcı (maker) ve en sonunda da sorgulayıcı/farklılaştırıcı (hacker) aşamalarına gelmelerini sağlamak.

21. yüzyılda her meslek grubu için, verimlilik ve rekabet adına artık dijital teknolojilerin kullanımı kaçınılmaz. Bunun temelini oluşturan programlama ise süratle ulusal eğitimin bir parçası olmalı. Bununla; okullara göstermelik ve ne olduğu anlaşılmayan bir ders olarak değil, her detayı planlanmış, öncelikle bu eğitimi verebilecek

öğretmenlerin bunu içselleştirmesiyle başlayacak ve yeni kuşağa ders izlemeyi bir külfet değil keyif haline getirecek uzun soluklu bir dönüşümden söz ediyoruz.

Kuşkusuz, bu zorlu ama ülkemiz adına fırsat olabilecek bir süreç. 15 yıl sonra mevcut mesleklerin yarısı yok olacak ve o dönemde popüler olacak mesleklerin de yarısını bugünden öngörmek zor. Ancak önümüzdeki döneme ilişkin herkesin üzerinde mutabık kaldığı yegane husus, dijitalleşme ve kodlama becerisinin tüm meslek kollarında rekabet ve verimliliği artıracağı yönünde. Bu yüzden dijitalleşme birikimi ve kodlama becerisi yüksek bir bireyin hangi mesleği seçerse seçsin, kariyerinde diğer meslektaşlarına göre önde yer alması kuvvetle muhtemel.

İşte şu anda okumakta olduğunuz kitap, sözünü ettiğimiz çocuklarımızı bu yeni yüzyıla hazırlayacak uzun sürecin ilk aşaması olan kodlamanın en basit ve eğlenceli yöntemlerinden biri olan Scratch programlama dilini eğlenceli bir üslupla ama sağlam bir temelle öğretmek amacıyla sevgili Hakan Ataş ve Bager Akbay tarafından hazırlandı.

Çocuklarını zamanın ruhuna uygun hazırlamayı planlayan anne-babalara; hem bu dönüşümü 12 yıldan beri mesleki olarak izleyen ve bunun eğitimini veren bir iletişimci, hem de bu yoldan daha önce geçen ve somut faydasını gören bir baba olarak bu kitabı ve devamında gelecek diğer kitap ve eğitimleri şiddetle tavsiye ediyorum. Umarım çok daha fazla çocuğumuz bu yollardan geçer ve bu yeni kuşak gençler gelecekte ülkemizi bizlerin düşlediğinden daha ileri noktalara götürür. Bizler de onların gururunu anne-baba ve öğretmenleri olarak hep birlikte yaşarız.

İsmail Hakkı Polat
Kadir Has Üniversitesi
Yeni Medya Bölümü
Öğretim Görevlisi

TEŞEKKÜR

Bu kitaba vesile olan tüm öğrencilere,

CoderDojo Türkiye'nin kurucuları Neşen Yücel ve Patrick Bosteels'e,

Fikirleriyle ve enerjileriyle destek olan değerli dostlarımız Sayat Samyeli ve Yalın Güçkıran'a,

Her tür projede desteklerini esirgemeyen Robotistan kurucularından Yasir Çiçek, İlge İpek, Yusuf Gündoğdu ve Sait Borlak'a,

Öğrenme ve öğretme arzularıyla dolu kodlamaogren.com ekibinden Kerim Karaköse, Hasan Esen, Yağmur Gür, Begüm Pişkin ve Emre Bodur'a,

Maker hareketinin Türkiye'de yayılmasını sağlayan herkese,

CoderDojo-İskele47 mentörleri, Cansu Yalçıner, Orhan 'aib' Kavrakoğlu, Rüya Yalçın, Yasemin Şahintürk, Murat İnan, Esra Çomu ve İmge Akbulut'a,

Kitabın baskılarında düzeltmeleriyle ve geri dönüşleriyle bize destek olan Şenol Yüksel öğretmenimize,

İstanbul Mini Maker Fuarı'nın küratörleri Halil Aksu, Ziya Bahtiyar, Gizem Renklidağ, Ekmel Ertan, Umut Erkal, İlker Vardarlı, Alper Kerpiççi, Ramazan Subaşı, Nebi Cihan Çankaya'ya,

Her tür etkinliğe mekânlarını açan ve destek olan İskele47 ekibinden Osman Koç, Selin Özçelik Mörth, Nagehan Kuralı Alan, Zeynep Nal Sezer'e,

Özgür Yazılım ve Açık Donanımın bu topraklarda yayılmasını sağlayan İstanbul Hackerspace ve Alternatif Bilişim Derneğine,

Kafa açan muhteşem sohbetleri için; Erdem Dilbaz, Engin Ayaz ve Barış Büyükakyol'a.

Maker Eğitim Hareketine destek veren herkese,

Buluş Biliş (BUBİ) Derneği'nden Elif Demirci ve Berkay Tunalı'ya,

Bu kitabın yazarlarının tanışmasına vesile olan Aynur Anıl Karamahmutoğlu'na ve küçük Masal'a,

Kitabın hazırlanmasında emeği geçen diğer tüm paydaşlara, özellikle Cevahir Demiryakan'a teşekkür ederiz.

Merhaba Genç!

Bu kitabı senin için yazdık ama sen bu satırları okumayıp, doğrudan konulara başlayacaksın, ki bizce de en doğrusu bu, izin verirsen senin yerine velilere ve öğretmenlere bir kaç noktadan bahsetmek istiyoruz. İstersen, tabii ki okuyabilirsin, ilgini çekebilir.

Sevgili Veli&Sevgili Öğretmen,

Bu kitabı elinize aldıysanız, kafanızda çocuklarımızın teknoloji ile arasının nasıl olması gerektiğine dair birçok soru olmalı diye düşünüyoruz. Teknoloji diğer birçok alanla benzer özellikler taşıyor. Örneğin, çocuğunuz mutfaktan çıkmıyor olabilir, ama bunu iyi veya kötü bir şey olarak tanımlamadan önce mutfakta ne yaptığını anlamamız gerekiyor. Sürekli bir şeyler mi yiyor, yoksa sürekli yemek mi yapıyor? Yemesi de kötü bir şey değil tabii ki ama yemek yapmayı da öğrenmesi gerekiyor. Hatta yemek yapmaktan keyif alan bir çocuğu desteklemek, ona iyi araçlar almak, metotları öğrenebileceği kaynaklara ve öğrenme/deneme tekniklerine yönlendirmek gereklidir. Arada bir aşçının yanında durması, yaptıkları hakkında gerçekçi yorumlar alabilmesi de çok önemli. Teknoloji konusunda hevesli gençlerin durumu da aslında aynı. Üretiyorlar mı? Tüketiyorlar mı? Asıl sorumuz bu.

Peki nedir bu iyi araç gereçler ve metotlar?

Teknolojik Araçlar

Yazılım yazabilmek insanın doğal ortamda yapabileceği bir şey değildir. Yazılım yazmayı öğrenmek için üst düzey bilişsel deneyimler yaşanması gereklidir. Programlama dilleri bu tarz bir iş için hazırlanmış standartlardır. Programlama dillerini öğrenmek ve öğretmek hep çok zor olmuştur, lakin bu konuda yapılan son çalışmalar işleri biraz değiştirdi. Siyah ekranda saatlerce yazı yazılan ortamlardan, Scratch gibi sürükle bırak şeklinde yazılan programlama dillerine gelmiş bulunuyoruz. Scratch bir tür görsel programlama dilidir (visual programming language). Bu tip dillerin en önemli özelliği yazım hatalarına (syntax error) izin vermemeleridir. Bunun dışında da görsel ve işitsel projelerin yapılması bu dillerde kolaylaştırılmıştır. Bu yüzden öğrenilmesi oldukça kolaydır.

Yazılım 1-2 yıllık yoğunlaştırılmış bir eğitimle çözülebilecek bir sorun değildir, çizim yapmaya benzer. Bir çocuk 2-3 yaşında çizim yapmaya başlarsa, çevresindeki nesneleri görüşü zamanla değişir ve çizim yapmadığında bile çizim yeteneğini geliştirmeye devam eder. Belirli motor becerileri ve deneyimleri kazandığında oldukça iyi çizmesi işten bile değildir. Bu yüzden programlama eğitimine 4-5 yaşında başlamak gerekmektedir. Bunun amacı, çocuğun hemen proje geliştirmesi değildir. Bu sistemlere aşina olması sağlanıp, günlük hayattaki mekanizmaları anlaması hedeflenmelidir. Küçük yaşta bu farkındalığı kazanan çocuklar için programlama öğrenmek çok daha kolay olacaktır.

Ayrıca öğrenmek için bir yazılım dilini seçerken, bu dilin bize uzun vadede yararlarına bakmak gerekmektedir. Bu yüzden belirli ticari amaçlarla üretilmiş ve belirli firmalara ait olan dilleri seçmek hatalı bir davranış olabilir. Bu firmalar, gelecekte kendi çıkarları gereği bu dilleri kapatabilir veya amaçlarını tamamen değiştirebilirler. Bu yüzden bizler kendi çocuklarımıza bir yazılım öğrettiğimizde bu yazılımın **özgür yazılım*** olmasını istiyoruz. Bu sayede bu dil herkese açık oluyor ve topluluklar tarafından yönetiliyor. Aynı şey donanım kullanırken de geçerlidir. Örneğin, bu kitapta anlatılan makey makey adlı ürün **açık donanım*** kategorisine giriyor. Açık donanım olan bir ürünü herkes üretebililiyor dolayısıyla hem fiyatı oldukça uygun oluyor, hem de çeşitliliği yüksek oluyor.

Yöntem

Madem çocuklarımızın geleceğin üreticileri olmalarını istiyoruz, bu konuda da bazı şeyleri netleştirmemiz gerekiyor. Özellikle büyük şehirlerde üretim yok denecek kadar az ve hızla azalmaya devam etmekte. Üretim mekânlarının yerlerini toplantı salonları aldı; bunu geri dönüştürmemiz lazım yoksa herkes pazarlamacı, taşımacı veya hizmet elemanı olacak; bu şekilde güzel yarınları hayal etmek pek mümkün değil. Üretim mekânlarını hatırlamaya çalışalım; bu tip mekânlar dağınıktır, bir mekânın derli toplu olması oranın tanıtım amaçlı kullanıldığını gösterir. Üretim mekânlarında birçok ufak tefek alet edavat ve birçok sarf malzemesi bulunur. İçerisi canlıdır, sürekli bir şeyler üretilir. Son yıllarda birçok eğitim kurumunun "üretim yapıyoruz" iddiasıyla birkaç pahalı makine alıp, odanın kullanım saatlerini sınırlayıp, makineler bozulmasın diye kimseye elletmediklerine maalesef şahit oluyoruz. Bir mekânın değeri orada çalışanlar ve o çalışanların ürettikleridir, içerideki pahalı ekipmanlar değil. Bunu unutmamalıyız.

- Çocuklar kopyalar, bozar, değiştirir. Hatta tüm insanlar böyle öğrenir. İnsan ustalaşana kadar kopya çeker, hatta büyük endüstriler bile bu kopyalama yöntemleri üzerine kuruludur (everything is a remix). Çocukların kopyalamasına izin ve fırsat veriniz.
- Bozmak bir aletin işlevi dışında kullanılmasıdır, tıpkı yoğurt kabından saksı yapmak gibidir. "Çocuğunuz hacker olmak zorunda" diyoruz, çünkü 20. yüzyıl bize endüstri standartlarını getirdi, bunların birçok yararı vardı ama en önemli zararı çeşitliliği azaltmasıydı. Bu yüzden size sunulan aletlerle yetinmek zorunda değilsiniz, onları bozmak, işlevi dışında kullanmak en önemli hakkınızdır.
- Hata, öğrenmenin en önemli unsurudur; lakin eğitim sistemi doğruları, başarıyı sever. Bırakın hata yapsınlar, başka türlü öğrenemezler. Bunu sizden özellikle talep etmedikleri sürece, onlara doğruyu göstermeyin. Bir insana öğrenmeyi öğretmek verilebilecek en güzel hediyedir.

Bu yazıda geçen konular sadece bizim fikirlerimiz değil, tartıştığımız birçok usta/uzman ile karar verdiğimiz ortak fikirler, bu yüzden buradaki fikirlerin sahipleri yok.

Bu kitabı verdiğimiz eğitimlerin ve ürettiğimiz projelerin arta kalan zamanlarında yazdık. Elimizden geldiğince basit ve yararlı olmasını hedefledik. Sizlerin bu kitaptan çok daha iyilerini yazacağınıza eminiz ve umarız en kısa zamanda bunu yaparsınız. Varsa hatalarımız, şimdiden affola.

'Eğer daha ilerisini görebildiysem bunun sebebi devlerin omuzları üzerinde durmamdır'.

Sevgilerimizle

Hakan Ataş, Bager Akbay

*** Özgür Yazılım:**

- Her türlü amaç için yazılımı çalıştırma özgürlüğü (özgürlük 0)
- Yazılımın nasıl çalıştığını inceleme ve kendi gereksinimleri doğrultusunda değiştirme özgürlüğü (özgürlük 1). Yazılım kaynak koduna erişim bunun için bir ön şarttır.
- Yeniden dağıtma ve toplumla paylaşma özgürlüğü (özgürlük 2).
- Yazılımı geliştirme ve gelişmiş haliyle topluma dağıtma özgürlüğü (özgürlük 3).

*** Açık Donanım:**

- Tam dokümante edilmiş.
- Gerekli yazılımlar özgür yazılım.
- Şeması ve parça listesi, yerleşim planları verilmiş.
- Üzerinde değişiklik yapılabilir.
- Aynısı yeniden üretilebilir ve satılabilir.
- Creative Commons ve diğer lisanlara sahip.

Kaynak: linux.org.tr

Eğitim, çocuklar ve teknoloji hakkındaki fikirlerimizi kökünden değiştiren
31 Temmuz 2016'da kaybettiğimiz Seymour Papert'a ithaf olunur.

İÇİNDEKİLER

1. Online Editor 1
   - Scratch Web Sitesine Kayıt Olma 4
   - Çevrimiçi Editörü Tanıyalım 7
   - Scratch Yardım Bölümü 9
   - Projelerin İçini İncelemek 10
   - Stüdyo Oluşturmak 11
2. Offline Editor 13
   - Offline Editörü Kuralım 14
   - Offline Editörü Tanıyalım 16
   - Kodlamaya Başlayalım 18
   - Başka bir Projeden Karakter Aktarma 19
3. Çizim Araçları ve Animasyon 25
   - Bitmap - Vektörel 27
   - Çizgisel (Vektörel) Grafik Nedir? 27
   - Scratch Çizim Araçlarını Tanıyalım 28
   - Çizim Yapalım 30
   - Birden Çok Kılık Oluşturmak 32

PROJELER

4. DJ Scratch 35
   - Sahnemizi Düzenleyelim 36
   - Ses Dosyası Ekleyelim 38
5. Pong 45
   - Raketimizi Oluşturalım 46
   - Topumuzu Oluşturalım 49
   - Topun Sekmesini Sağlayalım 52
6. Kovalamaca 57
   - Fare Okunu Takip Edelim 58
   - Kendi Zaman Sayacımızı Oluşturalım 60
7. İsmimizi Canlandıralım 65
   - Harflerimizi İçeri Aktaralım 66
   - Görsel Efektler Kullanalım 68
   - Geri Sayım 70
8. Quiz Uygulaması 77
   - Karakterlerimizi Seçelim 78
   - Sorulamızı Soralım 79
   - Otomatik 4 İşlem Soruları 82
9. Kaplumbağa Geometrisi 87
   - Düzgün Çokgenler Çizelim 88
   - Kaplumbağa ile Sanat 91

10. Kalem Sekmesi 99
Fare İmleciyle Çizim Yapmak 101
Koordinat Düzlemi 103
Kalem ile Harf Çizmek 104
İz Bırak Aracını Kullanmak 105
11. Bilye Yarışı 111
Bilye Yarışı 112
Bilyemiz Düzgün Hızlanıyor 114
12. Karşıya Geçme 119
Yolumuzu Oluşturalım 120
Yengeç Hareket Ediyor 122
Yengeç Karşıya Geçmeye Çalışıyor 125
Yaşam Ekleyelim 128
13. Zıplayan Kuş 137
Çalılık Ekleme ve Hareket 139
Engel Ekleme 141
Kuş ve Hareketi 143
Kanatlar 145
Skor Eklemeyi Unuttuk Sanmayın 146
14. Tuğla Oyunu 149
Pong Oyununa Yaşam Ekleyelim 150
Tuğlalarımızı Oluşturalım 151
Haber Gönderelim 153
Geri Sayım 156
15. Akvaryum 163
Akvaryum Seçelim 164
Akvaryuma Balıkları Atalım 165
Dalgıç Akvaryuma Giriyor 167
16. Hareket ve Ses Algılama 175
Scratch Üzerinde Web Kamera ile Kullanılabilecek Komutlar 176
Kamera ile Kediyi Hareket Ettirelim 177
Pico'lar Düşüyor 178
Pico'ları Kurtaralım 180
Ses ile Uçan Kuş Oynamak 182
17. Makey Makey 187
Makey Makey ile Neler Yapabilirim? 188
Nasıl Çalışıyor? 188
Makey Makey Kimin için? 188
Makey Makey Hangi Materyallerle Çalışır? 188

MAKER EĞİTİM KİTAPLIĞI

Bilişim Okur-yazarlığı

Bilgisayarın sosyal medya ve oyun dışında kullanıldığını gören bir gençlik.

Analitik düşünme

Düşüncelerini sıraya koyabilen ve uygulayabilen öğrenciler.

Takım Çalışması

Takım arkadaşlarıyla uyumlu çalışmanın önemini anlamış bir gençlik.

Paylaşma

Bilginin paylaştıkça çoğaldığını fark etmiş bir gençlik.

SCRATCH

M.I.T 21. yüzyıl Teknolojisi

Scratch M.I.T tarafından geliştirilmiş görsel bir programlama dilidir.

Yaratıcılık

Denemekten ve üretmekten çekinmeyen bir gençlik.

Disiplinler arası etkileşim

Tek derse bağlı değil, birçok dersi aynı anda potasında eritebilen öğrenciler.

Programlama

Teknolojiyi tüketen değil teknoloji üreten bir gençlik.

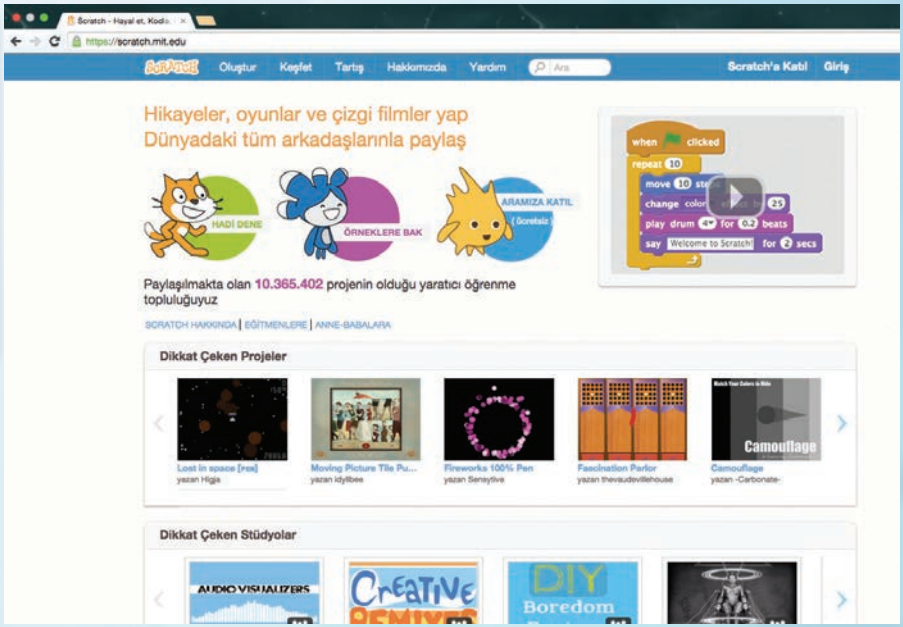
İşbirliği

Dünya üzerinde farklı kıtalar ve kişilerle işbirliği yapabilen öğrenciler.

Online Editor

Bu Aktivitede Neler Öğreneceğiz?

- Web sitesine kayıt olmayı,
- Yeni bir proje oluşturmayı,
- Var olan bir projeye katkıda bulunmayı.



BİLGİ

Scratch programlama dili, Logo programlama dili temel alınarak MIT tarafından 2007 yılında piyasaya sürülmüştür.

EK GÖREV

- Online olarak bir proje daha yapalım.
- Scratch web sitesinde beğendiğimiz bir uygulamaya katkı yapalım (remix).
- Kendi stüdyolarımızı oluşturalım ve beğendiğimiz projeleri eklemeye başlayalım.

Scratch programı online ve offline olarak 2 farklı şekilde çalıştırılabilir. Bu bölümde nasıl online çalıştıracağımızdan, bir sonraki bölümde ise offline editörü nasıl bilgisayarımıza yükleyeceğimizden ve giriş ekranlarından bahsedeceğiz.

Haydi başlayalım!

İnternette dolaşmak için kullandığımız tarayıcıyı açalım (google chrome, safari, mozilla firefox, internet explorer, opera ya da başka bir program) **https://scratch.mit.edu** adresine giriş yapalım. (Şekil-1)

Şekil 1

BİLGİ

Sayfanın dili Türkçe değilse en alttan dili değiştirebilirsiniz.

Öncelikle ekranın alt kısımlarına doğru gezinip neler olduğunu görüntülemenizi ve sağ üstte bulunan küçük tanıtım videosunu izlemenizi tavsiye ederiz.

Ekranın üst kısmında bulunan menüleri inceleyelim. (Şekil-2)

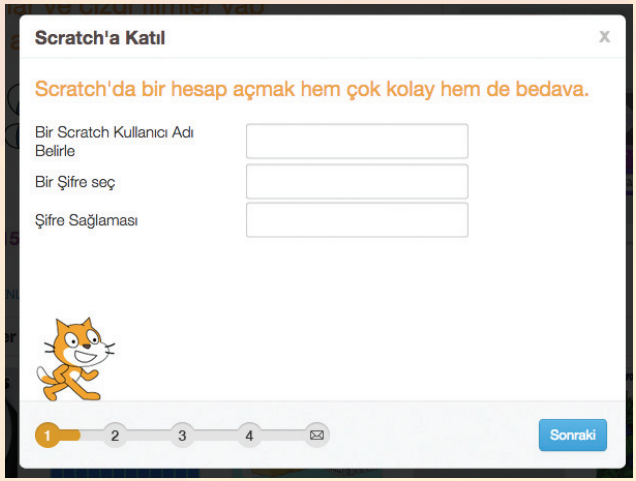
Şekil 2

1. Yeni bir Scratch dosyası oluşturmak için tıklamamız yeterlidir. Açılacak pencerede Scratch programına ait tüm özellikleri kullanabiliriz (Üye girişi yapmadığımız sürece online kaydetme özelliğini ve **ekleme** özelliğini kullanamayız).
2. Keşfet düğmesine tıkladığımızda Scratch üzerinde bulunan projelere göz atabilirsiniz. Belirli kategorilere ayrılan projeleri rastgele inceleyebileceğimiz gibi etiketlerine göre arama yapabiliriz.
3. Tartış düğmesine tıkladığınızda Scratch sayfasının forum bölümüne ulaşırız. Burada sormak istediğimiz ya da paylaşmak istediğimiz bir konunun daha önce sorulup sorulmadığını araştırabiliriz. Eğer daha önce konuşulmamışsa yeni bir başlık açabiliriz.
4. Hakkımızda kısmında Scratch hakkında genel bilgilere ulaşabilir, siteyi nasıl kullanabileceğimiz ve derslere nasıl adapte edebileceğimiz hakkında geniş bilgi edinebiliriz.
5. Yardım kısmında Scratch kılavuzlarını bulabilir, Scratch kartları ile takibi kolay küçük projeler yapabiliriz. Ayrıca bu bölümde video anlatım bağlantılarını da bulabiliriz. Sayfanın sağ tarafında ise Scratch hakkında bulunan kaynaklardan bazıları listelenmiştir.
6. Scratch web sitesinde genel olarak arama yapmamıza olanak sağlar.
7. Scratch web sitesine kayıt olmak için bu düğmeye tıklayalım.
8. Eğer Scratch sitesine daha önce kayıt olduysak bu bölümden giriş yapalım.

Scratch Web Sitesine Kayıt Olma

Scratch web sitesine kayıt olmak için, siteye ilk girdiğimizde sağ üstte bulunan **Scratch'e Katıl** düğmesine tıklayalım. Ardından oldukça basit bir seviyede düzenlenmiş 3 basit kademede, yaklaştık bir dakikada, siteye kayıt olabiliriz.

1. Daha önce kullanılmamış bir kullanıcı adı ve şifre seçmemiz gerekmektedir. (Şekil-3)



Şekil 3

2. 2. aşamada; doğum tarihimizi, cinsiyetimizi ve ülkemizi girmemiz için bir form çıkıyor. (Şekil-4)

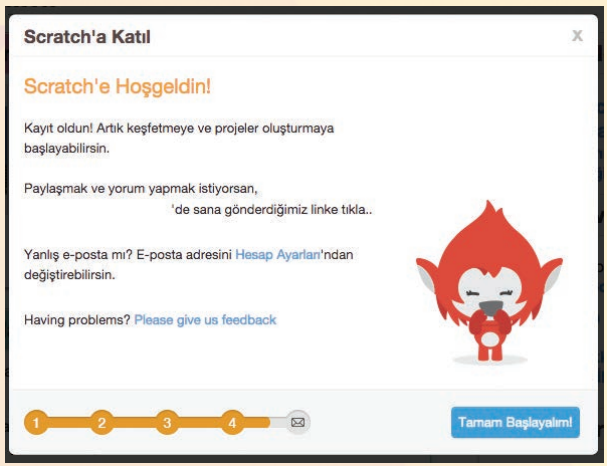
Şekil 4

3. Son aşamada ise, bizden e-posta adresimiz isteniyor. Hesabımızı onaylamak için veya ileride şifremizi unutursak tekrardan alabilmek için, kullandığımız bir e-posta adresi girmeliyiz. (Şekil-5)

Kendi adınıza açılmış bir e-posta hesabınız varsa onu kullanabilirsiniz.

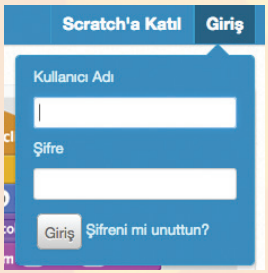
Şekil 5

4. Ve artık Scratch ailesine katıldık. Scratch üzerinde online yaptığımız işlemleri artık kaydedebilir, diğer projelerin içeriğini görebildiğimiz gibi onlara eklemeler de yapabiliriz. (Şekil-6)



Şekil 6

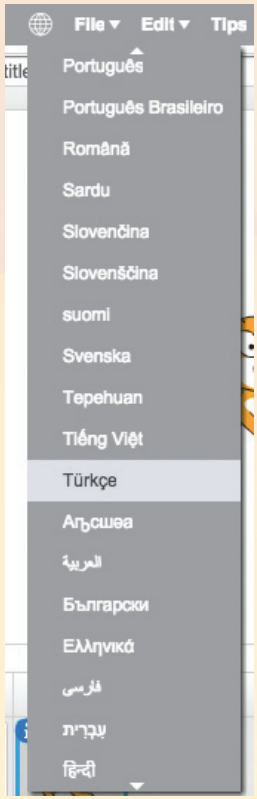
Kayıt işleminin ardından siteye giriş yapalım. Sağ üstte bulunan **Giriş** düğmesine tıklayıp, açılan küçük pencereye belirlediğimiz kullanıcı adı ve şifreyi girelim. (Şekil-7)



Şekil 7

Hesabınızı ortak kullanılan bir bilgisayardan açtıysanız ders sonunda hesabınızdan çıkış yapmayı unutmayın.

Web sitesinde özellikle projeler arasında biraz dolaşmamız ve yapılmış olan projeleri incelememiz yararımıza olacaktır. Kendi projemizi oluşturmak için yukarıda belirttiğimiz **Oluştur** sekmesine tıklayarak Scratch editörümüzü açabiliriz. Eğer editörün dili Türkçe değilse, sol üstte bulunan **dünya** ikonuna tıklayarak Türkçe seçeneğini seçebiliriz. (Şekil-8)

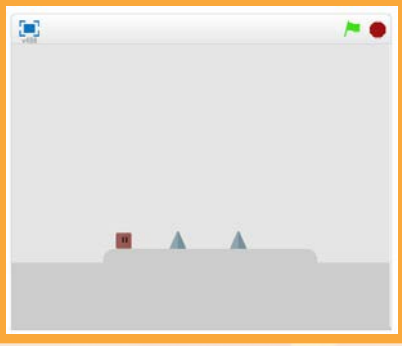


Şekil 8

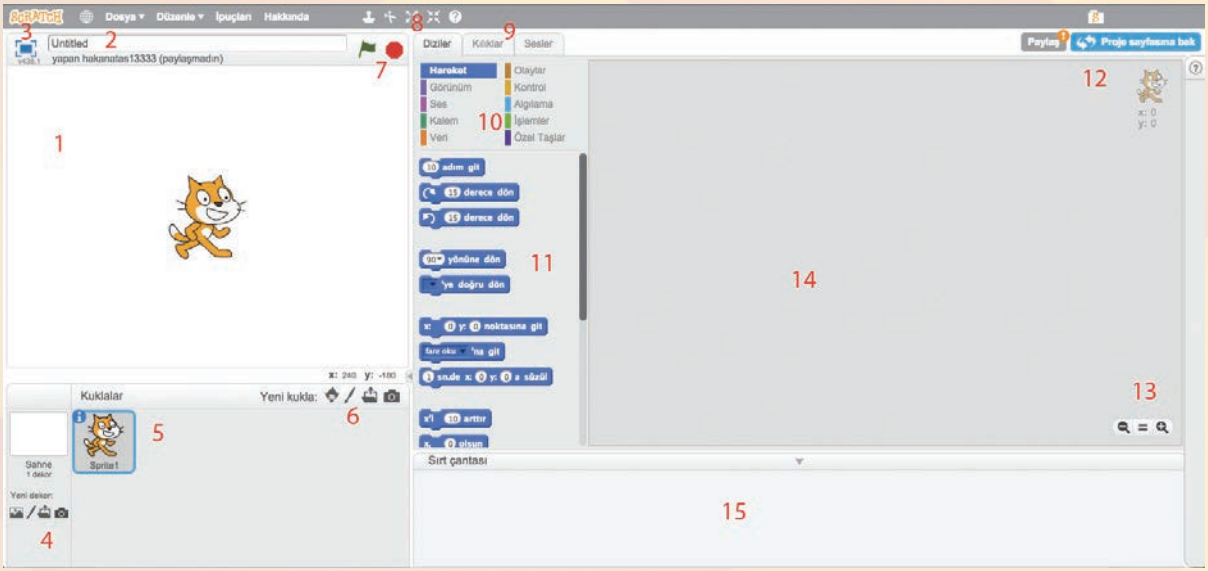
TAVSİYE PROJELER:

https://scratch.mit.edu/projects/10045024/

https://scratch.mit.edu/projects/94868644



Çevrimiçi Editörü Tanıyalım



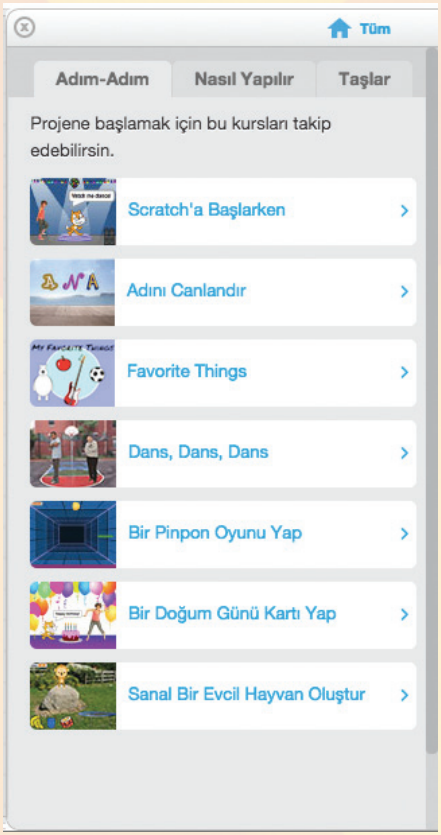
1. Uygulamamızın sahne alanıdır. Uygulama içerisinde yaptığımız tüm değişiklikleri bu alanda görüntüleyebiliriz.
2. Uygulamaya adını verdiğimiz bölümdür.
3. Uygulamayı tam ekran yapabileceğimiz komut düğmesidir.
4. Arka plan bölümüdür. Uygulamanız içinde yer alan arka planları (dekorları) belirleyebileceğimiz bölümdür. Uygulamanıza ait arka planları Scratch kütüphanesinden seçebilir, kendimiz çizebilir, bilgisayarımızdan herhangi bir resim aktarabilir ya da web kameramızdan içeri aktarabiliriz.
5. Karakterlerin sergilendiği alandır. Karakterlerin sol üstünde bulunan **i** sembolünü tıklayarak karaktere yeni bir isim verebilir ve dönüş izinlerini değiştirebiliriz.
6. Uygulamamızın içine Scratch kütüphanesinden, kendimiz çizerek, bilgisayarımızdan içeri aktararak veya web kamerasından karakter aktarabiliriz.
7. Yeşil bayrak, genel olarak uygulamalarımıza başlamak için kullanılır. Kırmızı düğme ise uygulamadaki tüm komutların durmasını sağlar.
8. Bu bölümde bulunan düğmeler sırasıyla bir bloğun kopyasını çıkartmak, kesmek, büyütmek, küçültmek ya da istenilen komut düğmesi hakkında yardım almak için kullanılabilir.

9. Bu bölümde komut bloklarının bulunduğu **diziler** sekmesi, kılıkların yer aldığı **kostüm** sekmesi ve karaktere ait ses dosyalarının bulunduğu **ses** sekmesi bulunmaktadır.
10. Sadece diziler sekmesi altında görebileceğimiz bir alandır. Bu bölümde kod bloklarının kategorileri yer alır.
11. Betikleme alanına sürükleyeceğimiz kod bloklarının bulunduğu alandır. Bu kod bloklarını üstlerine çift tıklayarak deneyebiliriz.
12. Bu bölümde var olan komut düğmeleri sadece online editörde bulunur. Bu bölümde bulunan Paylaş düğmesi ile, yaptığımız projeyi paylaşabiliriz. Eğer başkasının projesinin içindeysek, bu bölümde **Ekleme** düğmesini görürüz. Sağ üstte bulunan **Proje Sayfasına Bak** düğmesi ile, yaptığımız projelerin sayfasına ulaşabiliriz.
13. Yakınlaştırma düğmeleriyle betikleme alanına yakınlaşabilir ya da uzaklaşabiliriz.
14. Bu alan kod bloklarını birbirine takarak betiklerini yazabileceğimiz alandır.
15. Bu bölüm sırt çantası alanıdır. Bu bölüm sadece online editörde bulunur. Başka uygulamalarda gördüğümüz karakter, kod bloğu veya ses gibi beğendiğimiz tüm nesneleri buraya sürükleyerek ileride başka projelerimizde kullanabiliriz.

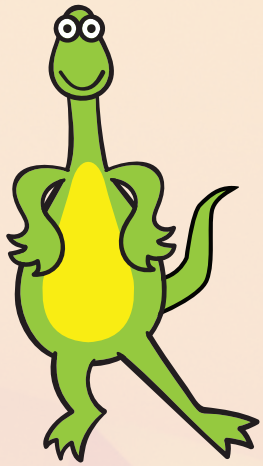
Scratch Yardım Bölümü

Editörün en sağında bulunan soru işaretli bölüme tıklarsak Scratch yardım bölümü açılacaktır. (Şekil-10) Bu bölümde 3 ayrı sekme vardır:

1. **Adım-Adım:** Bu alanda öğreticileri adım adım takip ederek yapabileceğimiz birbirinden güzel 7 tane proje bulunmaktadır.
2. **Nasıl Yapılır?:** Bir efekt, animasyon, hikâye canlandırma ya da bir oyun yapmak istiyoruz ama; örneğin "Köşeye gelince nasıl sekecek?" ya da "İki cismin birbirine çarptığını nasıl anlayacağım?" gibi sorularımız var. Bu bölümde, bu soruların cevaplarını bulabiliriz.
3. **Taşlar:** Bu sekmenin altında Scratch'in içinde bulunan tüm kod blokları hakkında açıklamalar ve küçük örnekler bulabiliriz.

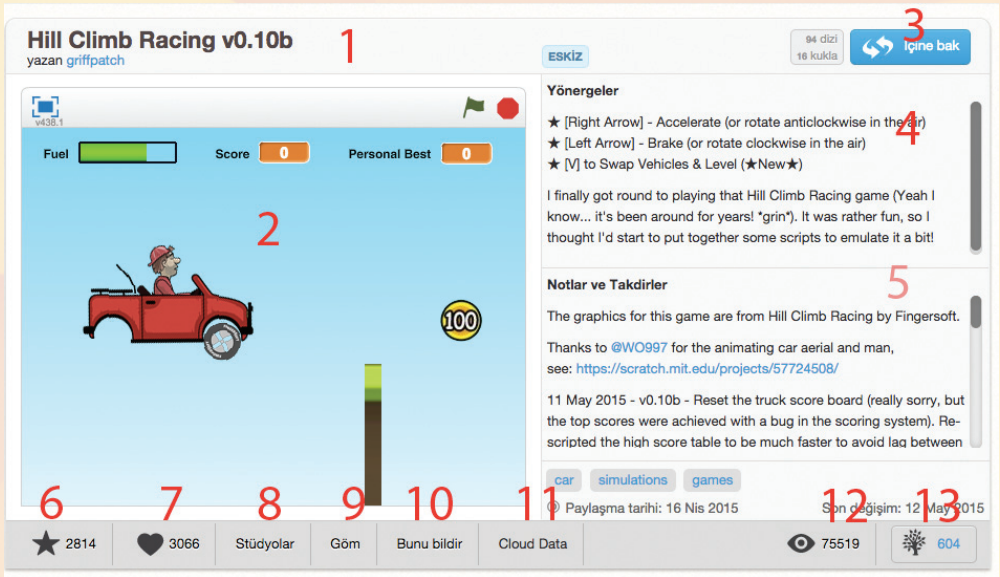


Şekil 10



Projelerin İçini İncelemek

Scratch üzerinde yapılan ve paylaşılan milyonlarca projenin herhangi birini seçtiğimizde; içindeki tüm kod bloklarına, karakterlere ve ses dosyalarına ulaşma imkânımız var. Üstelik bu çok kolay. Öncelikle, incelemek istediğimiz projeyi seçelim. Örneğin, **https://scratch.mit.edu/projects/56667360/** projesi oldukça eğlenceli. Verdiğimiz bağlantıda bulunan projeyi ya da kendi istediğimiz herhangi bir projeyi açalım. Proje sayfasına girdiğimizde karşımıza şöyle bir ekran çıkacaktır: (Şekil-11)



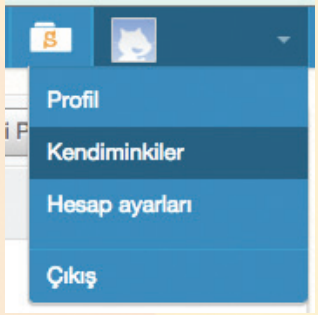
Şekil 11

1. Projenin adının ve kimin tarafından geliştirildiğinin görüntülendiği bölümdür.
2. Projenin kendisini görüntüleyebildiğimiz alandır. Tam ekran yaparak, uygulamanın keyfini çıkarabiliriz :)
3. Bu düğmeye tıkladığımızda; projenin kaynak kodlarını, kullanılan karakter ve ses dosyalarını görüntüleyebiliriz.
4. Projeyi geliştiren kişinin, oyunun nasıl oynanacağı veya uygulamanın nasıl kullanılacağı hakkında yönergelerinin bulunduğu alandır.
5. Projede kullanılan kaynaklar, yapılan işler ve versiyonlarla ilgili bilgilerin bulunduğu alandır.
6. Projeyi kaç kişinin gözdelerine eklediğinin belirtildiği alandır. Ayrıca, bu alana tıklayarak projeyi kendi gözdelerimize ekleyebiliriz.
7. Projeyi kaç kişinin beğendiğinin görüntülendiği alandır. Siz de bu düğmeye tıklayarak projeyi beğenebilirsiniz.

8. Projeyi daha önce oluşturduğumuz stüdyolardan birine kaydedebileceğimiz alandır.
9. Projeyi bir web sitesinin içine gömmek için gerekli kodların verildiği alandır.
10. Eğer bir projede şikayette bulunacağımız bir alan varsa (Örneğin, bir projemizden kaynağı belirtilmeden alınan bir resim gibi), bu bölümden bunu Scratch takımına bildirebiliriz.
11. Bulut verilerin geçmişini görüntüleyebileceğimiz alandır.
12. Projenin kaç kişi tarafından görüntülendiğinin belirtildiği alandır.
13. Bu projeyi ilham alarak, kaç kişinin bu projeye eklenti yaptığının ve atıfta bulunduğunun belirtildiği alandır. Bu düğmeye tıklarsak bu ağacı görüntüleyebiliriz.

Stüdyo Oluşturmak

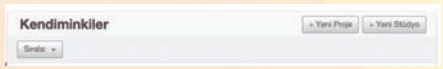
Scratch web sitesi üzerinde bir stüdyo, birçok projeyi bir sayfada görüntüleyebileceğimiz bir alandır. Bir stüdyo oluşturmak için, siteye giriş yapmamız ve sağ üstte adımızın bulunduğu bölümden **Kendiminkiler** bölümünü seçmeliyiz. (Şekil-12)



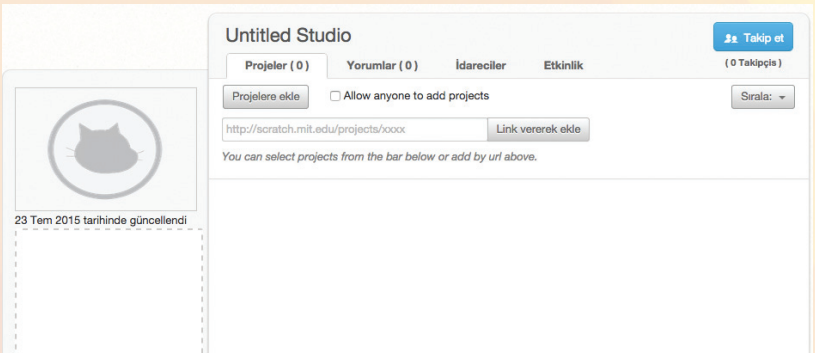
Şekil 12

Bu bölüme girdiğimizde, sağ üstte bulunan menülerden **+Yeni Stüdyo** düğmesine tıklayalım. (Şekil -13)

Şekil 13



Yeni stüdyomuza bir isim verebilir ve "Proje ekle" düğmesinden eklemek istediğimiz projenin URL adresini girerek bu stüdyo altında toplamak istediğimiz projelerimizi ekleyebiliriz. (Şekil-14)

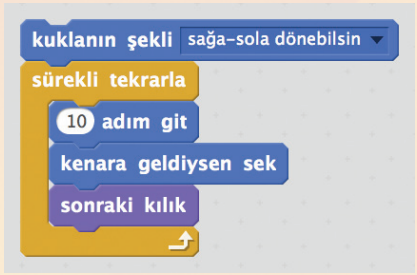


Şekil 14

ÇALIŞMA SAYFASI

1. **https://scratch.mit.edu/hoc/** adresinde bulunan 3 etkinlikten en az birini yapalım.
2. scratch.mit.edu adresine giriş yapalım ve en az 3 tane projenin kodlarını inceleyelim.
3. scratch.mit.edu adresinde, projeler arasında gezinirken beğendiğimiz karakterleri sırt çantasına atalım.
4. İstediğimiz bir konu hakkında bir stüdyo oluşturalım ve web adresinde gezinip bu stüdyoya projeler ekleyelim.
5. Yeni bir proje açıp aşağıdaki kod bloklarını deneyelim ve aralarındaki farkı görelim.

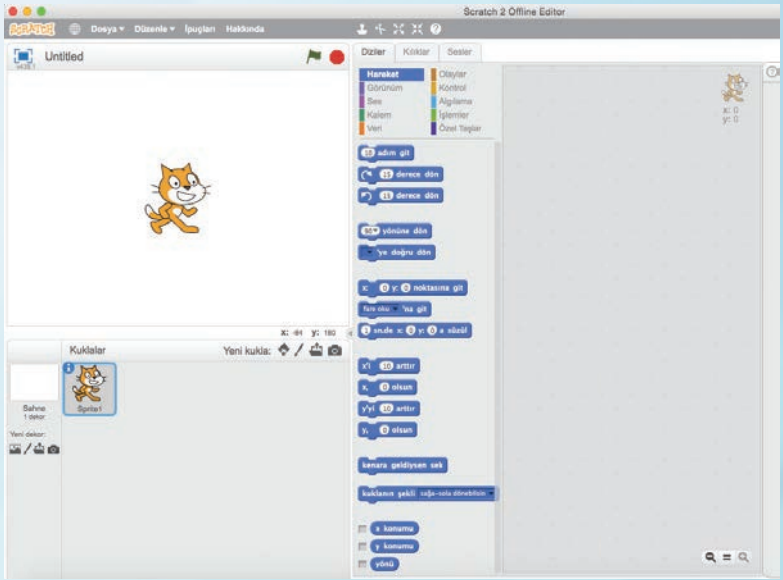
6. Aşağıdaki kod bloklarını deneyip aralarındaki farkı görelim.



Offline Editor

Bu Aktivitede Neler Öğreneceğiz?

- Scratch Offline editörünü kullanmayı,
- Dosyalarımızı kaydetmeyi,
- Kod bloklarını kullanmayı,
- İçeriye karakter aktarmayı,
- Klavye tuşlarını kullanmayı.



BİLGİ

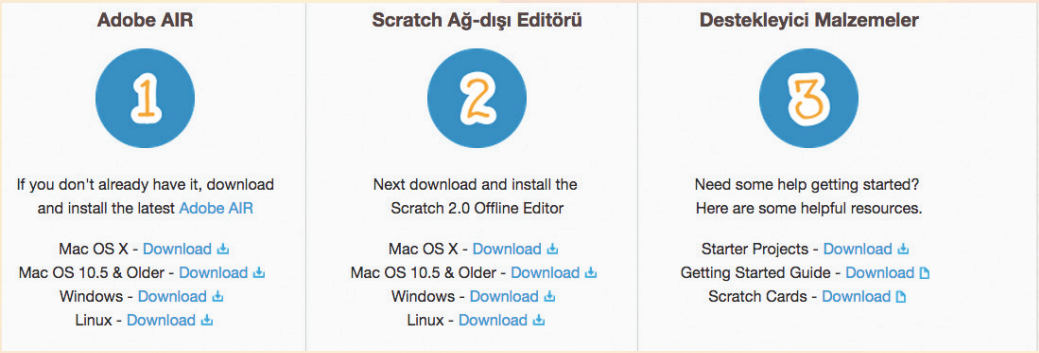
Scratch Offline Editör ile web sitesinde uygulama paylaşmak çok kolay. Tek yapmamız gereken Dosya menüsünün altından **Web sitesinde Paylaş** komutunu seçmek.

EK GÖREV

- Scratch kütüphanesinde bulunan diğer karakterlerle animasyon yapmayı deneyelim.
- Uygulamalarımıza kendi sesimizi eklemeyi deneyelim.
- Başka bir arkadaşımızın dosyasını kendi bilgisayarımıza aktarıp üzerinde değişiklik yapalım.

Offline Editörü Kuralım

Scratch offline editörünü kurmak oldukça kolaydır. Öncelikle **https://scratch.mit.edu/scratch2download/** adresine gidelim. (Şekil-1)



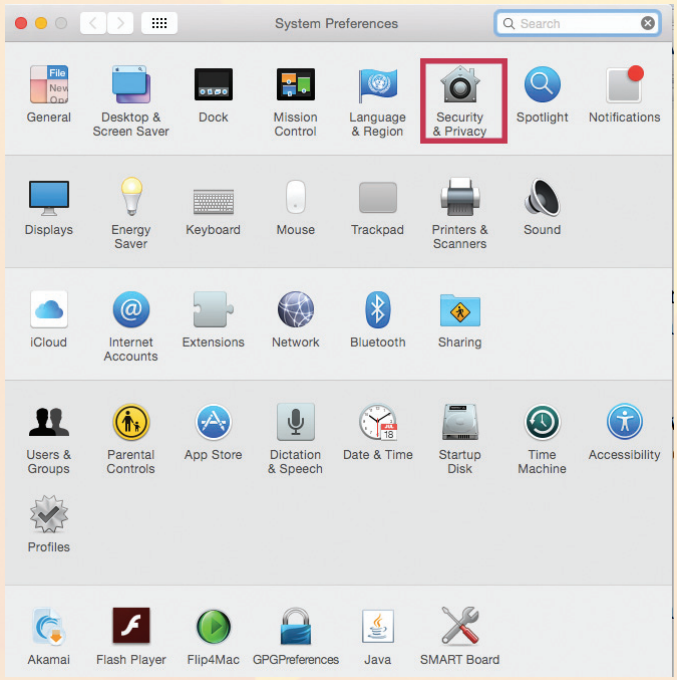
Şekil 1

Scratch programını kurabilmek için öncelikle 1 numaralı Adobe AIR programını yüklemeliyiz. Buradan, kullandığımız bilgisayara uygun olan Adobe AIR uygulamasını seçelim ve bilgisayarımıza indirelim. Bu programı kurduktan sonra, Scratch offline editörü indirip kurabiliriz.

2 numaralı alandan bilgisayarımız için uygun olan editörü indirelim. Sonrasında tek yapmamız gereken, programı kurmak. Uygulama için özel bir ayar gerekmemektedir.

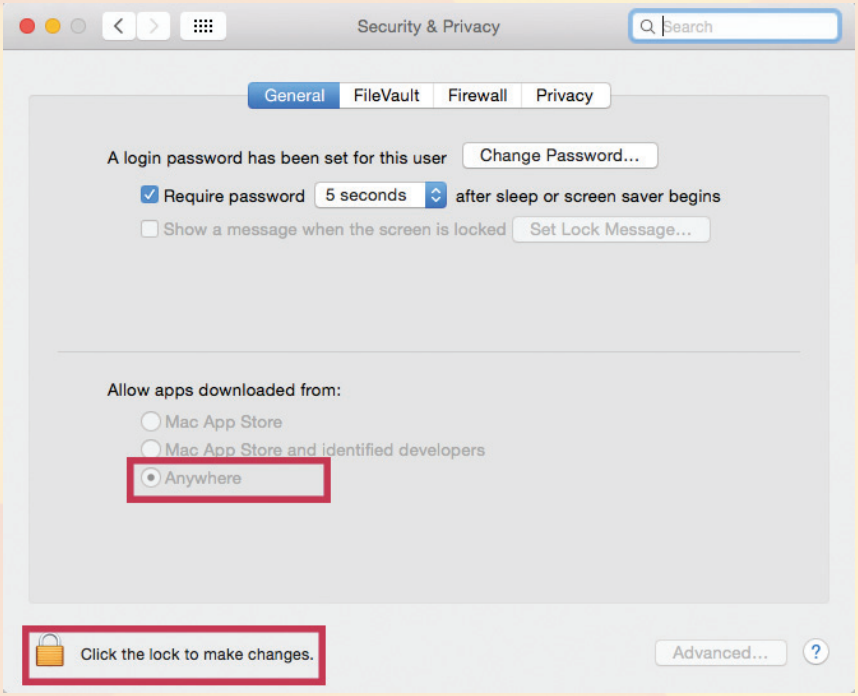
+ BİLGİ

Mac kullanıcıları için hatırlatma:
Uygulamayı dışarıdan indirdiğimiz için güvenlik ayarlarını değiştirmemiz gerekebilir. Bunun için Ayarlar içerisinden Güvenlik ve Gizlilik ayarlarını seçelim. (Şekil-2)



Şekil 2

Güvenlik ve Gizlilik ayarlarını seçtiğimizde çıkan ekranda **uygulamalara izin ver** sekmesinde "Herhangi bir yerden" düğmesini seçelim. Bu seçimin aktif olması için, sol altta bulunan kilit düğmesine basıp ardından şifremizi girmeliyiz. (Şekil-3)

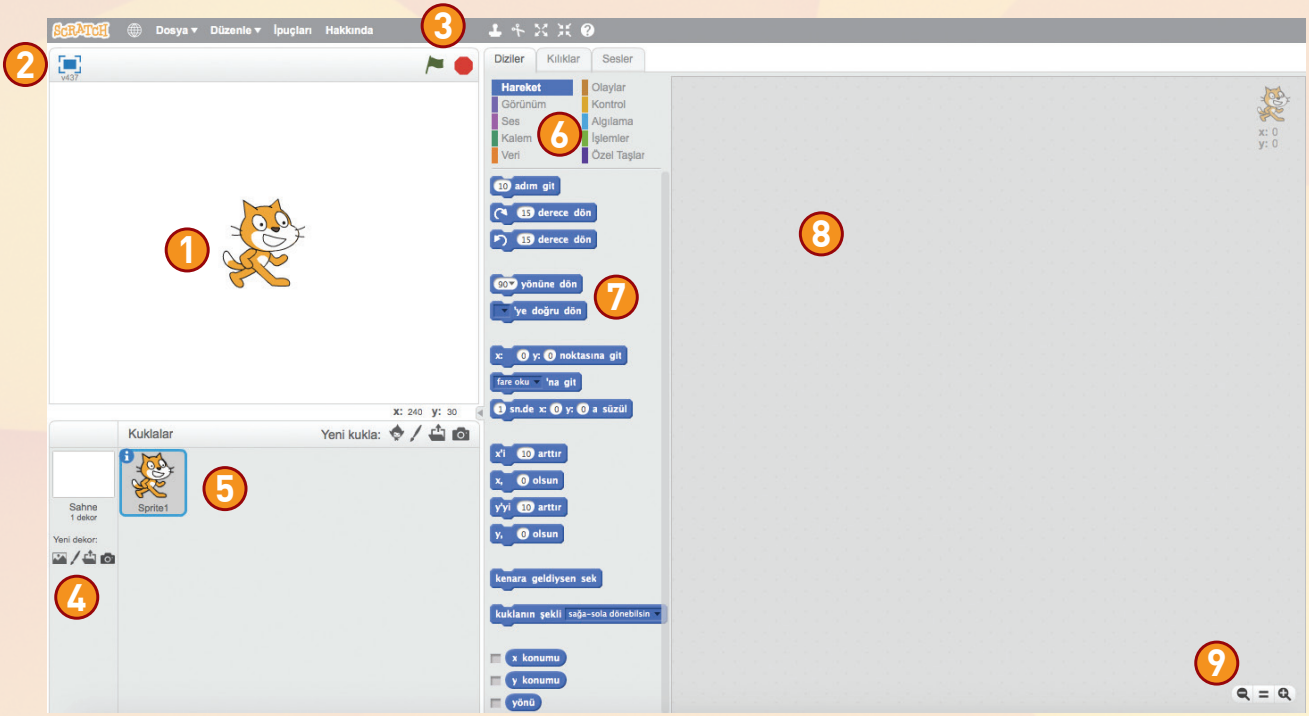


Şekil 3

BİLGİ

Windows İşletim Sistemi üzerinde kurulum yaparken aynı sırayı takip ederek kurulum yapabilirsiniz.

Offline Editörü Tanıyalım



1. Sahne: Uygulamamızın görüntülendiği alandır.

2. Tam Ekran: Uygulamamızı tam ekran yapmamızı sağlar.

3. Yeşil-Kırmızı Düğme: Yeşil bayrak genel olarak uygulamayı başlatır, kırmızı düğme ise uygulamayı tamamen durdurur.

4. Arka plan: Bu bölümde, sahnemizin arka planını düzenleyebiliriz. Arka planımızı Scratch kütüphanemizden seçebilir, kendimiz çizebilir, bilgisayarımızdaki bir resmi buraya aktarabilir veya web kamerasından çekim yapabiliriz.

5. Karakter Alanı: Uygulama içinde yer alan karakterlerin bulunduğu alandır. Bu alan içinde, karakterlerin üstlerine tıklayıp sol üstündeki mavi **i** ikonunu seçerek karakterimizin adını ve dönüş izinlerini değiştirebiliriz.

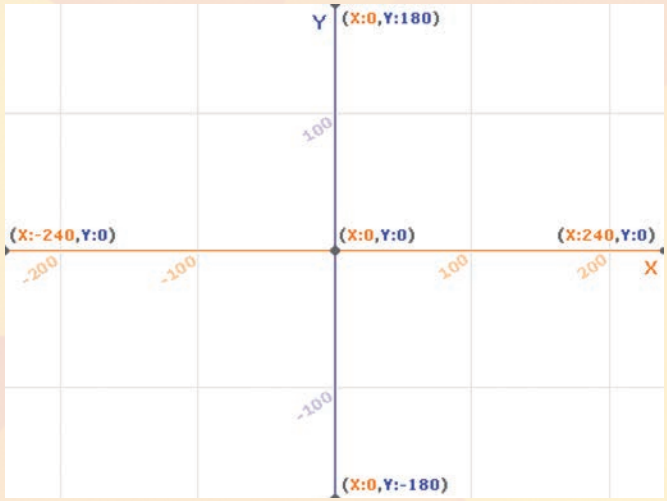
6. Sekmeler: Bu alana bir karakterin kod blokları, kostümleri ve ses dosyaları yüklenebilir.

7. Kod Blokları: Betikleme alanına sürüklenebilen kod bloklarının bulunduğu alandır. Kod bloklarına çift tıklayarak kod bloğunu deneyebiliriz.

8. Betikleme Alanı: Kod bloklarının yazılacağı (sürüklenip bırakılacağı) alandır. Her karakterin betikleme alanı farklıdır.

9. Yakınlaştırma: Betikleme alanını yakınlaştırıp uzaklaşırabileceğimiz komut düğmeleridir.

Scratch; x ekseninde (sağ-sol) -240, +240 aralığında yani 480 piksel genişliğinde, y ekseninde ise -180, +180 aralığında yani 360 piksel genişliğindedir.



BİLGİ

Herhangi bir kod bloğunu denemek için üzerine çift tıklamamız yeterlidir.

Bu bölümdeki uygulamamızı yapmaya başlayalım. Önce, klavyemizin yön tuşlarını kullanarak kedimizi hareket ettirelim.

BİLGİ KUTUCUĞU	
x'i 10 arttır	Sağa gitmek için.
x'i -10 arttır	Sola gitmek için.
y'yi 10 arttır	Yukarı gitmek için.
y'yi -10 arttır	Aşağı gitmek için.

Kodlamaya Başlayalım

1. Kodlamaya başlamadan önce, Kedi karakterimizin seçili olduğundan emin olalım. Klavyede bastığımız bir tuşu anlamak için 2 yolumuz var: Olaylar ve Algılama sekmeleri. Biz bu bölümde "Olaylar" sekmesini kullanacağız. (Şekil-4)

Şekil 4

2. Olaylar sekmesinin altında bulunan "**Boşluk tuşu basılınca**" kod bloğunu kodlama bölümüne sürükleyip bırakalım. (Şekil-5)

Şekil 5

3. Boşluk kelimesinin yanında bulunan siyah küçük oku tıklayınca çıkan seçeneklerden yön tuşlarını seçelim. (Şekil-6)

Şekil 6

4. 4 ayrı kod bloğunu sürükleyip yukarı ok, aşağı ok, sağ ok ve sol ok yön tuşlarını seçelim. (Şekil-7)

Şekil 7

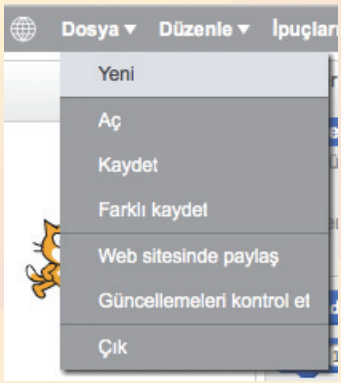
5. Şimdi, kedimizi hareket ettirmek için yön tuşlarıyla hareket bloklarını bir araya getirelim. Artık yön tuşlarını deneyebiliriz. (Şekil-8)

Şekil 8

6. Burada 10 diye gözüken sayıyı değiştirerek kedimizin hızının değiştiğini gözlemleyebiliriz. (Şekil-9)

Şekil 9

7. Kedimiz sağa sola giderken ayaklarını da hareket ettirelim mi? Öncelikle, yeni bir proje dosyası açalım. Nasıl mı yapıyorduk? (Şekil-10)



Şekil 10

Başka bir Projeden Karakter Aktarma

Açtığımız bu yeni projede, yine Scratch kedisini kullanacağız ama bu sefer farklı bir kedi kullanacağız. Şimdi, bir projedeki herhangi bir karakter nasıl içeri aktarılıyor öğrenelim. Öncelikle, istediğimiz karakteri bulmalıyız. **https://scratch.mit.edu/projects/69765034** adresine girelim (bu projenin orijinal hali **https://scratch.mit.edu/projects/10045024/** adresindedir). Eğer kullanıcı girişi yapmadıysak kullanıcı girişimizi yapalım.

+ BİLGİ

Eğer kullanıcı girişi yapmadıysanız "**içine bak**" düğmesini göremezsiniz!

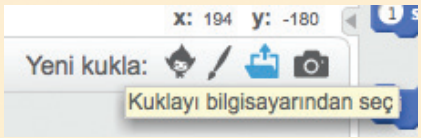
Ekranın sağ üst köşesinde "**içine bak**" düğmesini bulup tıklayalım. (Şekil-11)

Şekil 11

Projenin içinde girdiğimizde sol altta bulunan kedi karakterine sağ tıklayıp "**bilgisayarıma kaydet**" komutunu seçelim. Ardından, istediğimiz ismi verip bilgisayarımıza kaydedelim. (Şekil-12)

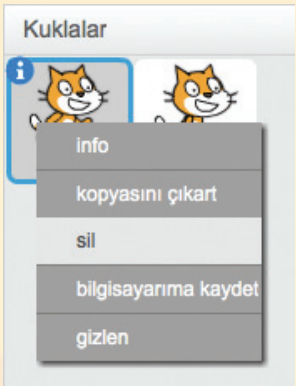
Şekil 12

Bilgisayarımızda yüklü olan Scratch programını açalım. Biraz önce bilgisayarımıza kaydettiğimiz "Scratch Kedisi"ni programımızın içine aktaralım. Bunun için, Karakter bölümünde bulunan "**Kuklayı bilgisayarımdan seç**" düğmesine tıklayalım. (Şekil-13)



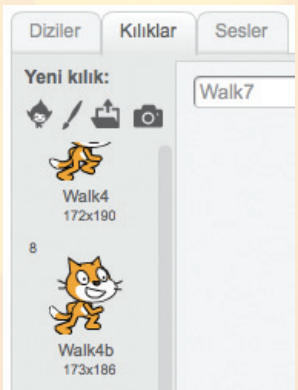
Şekil 13

Dosyayı kaydettiğiniz yeri işaretleyip tıkladığımızda kedi içeri aktarılacaktır. Şimdi iki tane Scratch kedimiz oldu. Eski Scratch kedimizi silelim. Silmek istediğimiz kediye sağ tıklayarak "Sil" komutuna tıklayalım. (Şekil-14)



Şekil 14

Kedimizi biraz harekete geçirme zamanı! Bir animasyon ya da çizgi film, peşpeşe resim karelerinin görüntülenmesiyle oluşur. Burada bizim için kedinin 16 tane ayrı ayrı çizilmiş yürüme hareketi var. Bunları görmek için orta bölümde bulunan "**Kılıklar**" düğmesine tıklayabiliriz. (Şekil-15)



Şekil 15

Bizim yapacağımız şey, kedinin bu kılıklarını peş peşe getirmek. Klavyenin sağ ok tuşuna bastığımızda kedinin biraz ileri gidip adımlarını hareket ettirmesini istiyoruz. Bunun için kullanacağımız bloklar yandakilerdir. (Şekil-16)

Şekil 16

İstediğimiz şey "Sağ Ok" tuşuna basıldığında kedinin 10 birim sağa gidip 16 defa (bu sayıyı değiştirebilirsiniz) sonraki kılığa geçmesi. (Şekil-17)

Şekil 17

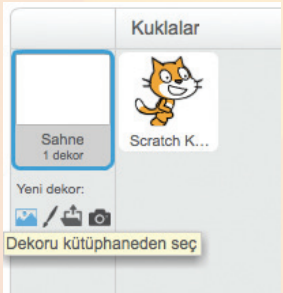
"Sol Ok" tuşuna basılınca benzer şekilde 10 birim sola gidip 16 defa (bu sayısı değiştirebilirsiniz) sonraki kılığa geçmesi. (Şekil-18)

Şekil 18

Klavyenin sağ ve sol yön tuşlarıyla denemelerimizi yapabiliriz. Yaptığımız bu uygulamayı kaydedelim. "Dosya" menüsünün altındaki "Kaydet" seçeneğine tıklayıp bir isim verdikten sonra dosyamızı kaydedebiliriz. (Şekil-19)

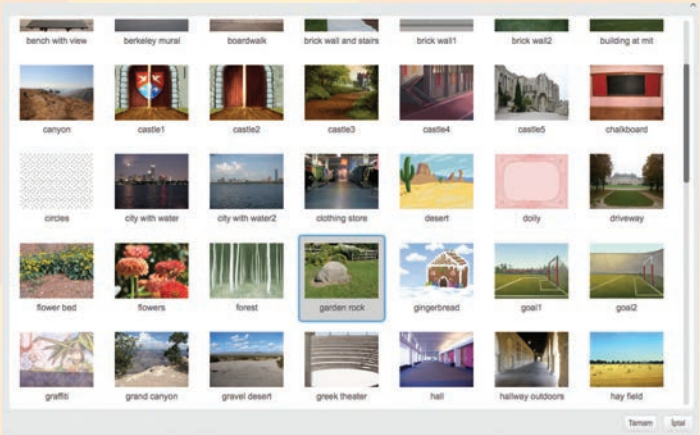
Şekil 19

Dilerseniz şimdi, arka planımızda değişiklik yapalım. İnternetten bulduğumuz bir görseli ekleyebilir veya Scratch kütüphanesinde hazır gelen bir arka planı kullanabiliriz. İlk önce, kütüphaneden bir arka plan seçelim. Bunun için sol altta bulunan sahne bölümündeki "**Dekoru kütüphaneden seç**" komutuna tıklayalım.
(Şekil-20)



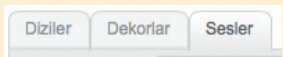
Şekil 20

Çıkan pencereden, istediğimiz bir arka planı seçip sağ altta bulunan Tamam düğmesine basalım. (Şekil-21)



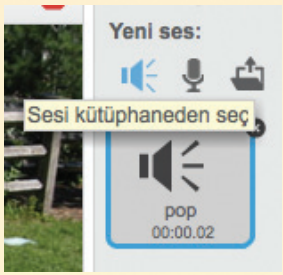
Şekil 21

Son olarak yürürken kedimizin "Miyav" sesi çıkarmasını ve sürekli olarak renk değiştirmesini sağlayacağız. Scratch programına ses eklemek için, orta bölümde bulunan "sesler" sekmesine tıklayalım. (Şekil-22)



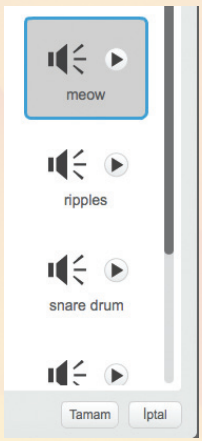
Şekil 22

Bilgisayarımızda bulunan bir sesi içeri aktarabileceğimiz gibi, bilgisayarımızın mikrofonu olması durumunda sesimizi kaydedebiliriz. Biz şimdilik kütüphaneden bir ses alacağız. (Şekil-23)



Şekil 23

Açılan pencereden "Meow" sesini seçelim. (Şekil-24)

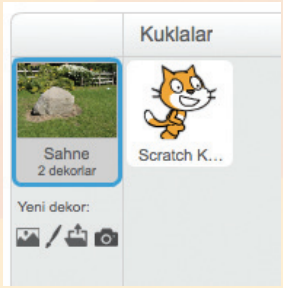


Şekil 24

Scratch web sitesi üzerinde Mayıs 2016 itibariyle **77.202.839** adet yorum yapılmıştır.

Şimdi sağ ve sol ok tuşlarına bastığımızda bir kez "Meow" sesini duyalım. Bunun için orta bölümde bulunan **Diziler** sekmesine geri gelip kedi karakterini tıklayalım. (Şekil-25)

Eğer şekildeki gibi, sahne seçili ise kod alanı boş olacaktır.



Şekil 25

Kedimize ait kodları görebilmemiz için kedi seçili olmalıdır. (Şekil-26)

Şekil 26

Kedimiz hareket ederken ses eklemek için Diziler sekmesinin altında bulunan Ses bölümüne gelelim. (Şekil-27)

Şekil 27

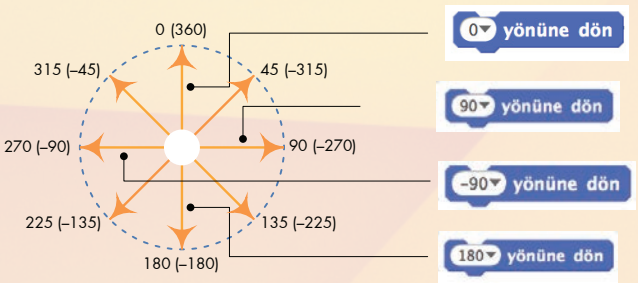
Buradan "**pop sesini çal**" komutunu sürükleyip çıkan menüden "Meow" sesini seçelim. Sol ok yönü için de "Meow" sesi ekleme bölümünü tekrarlayalım. (Şekil-28)

Şekil 28

ÇALIŞMA SAYFASI

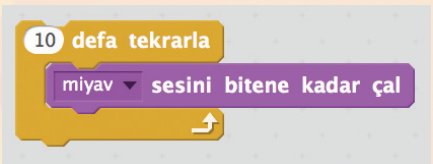
1. Hareket sekmesinin altında bulunan **Yönüne Dön** kod bloğunu inceleyelim.

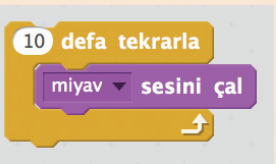
2. Scratch üzerindeki yönler şekildeki gibidir. İnceleyelim.



3. Aşağıdaki kod bloklarını uygulayıp deneyelim.

4. Aşağıdaki kod bloklarını deneyelim.

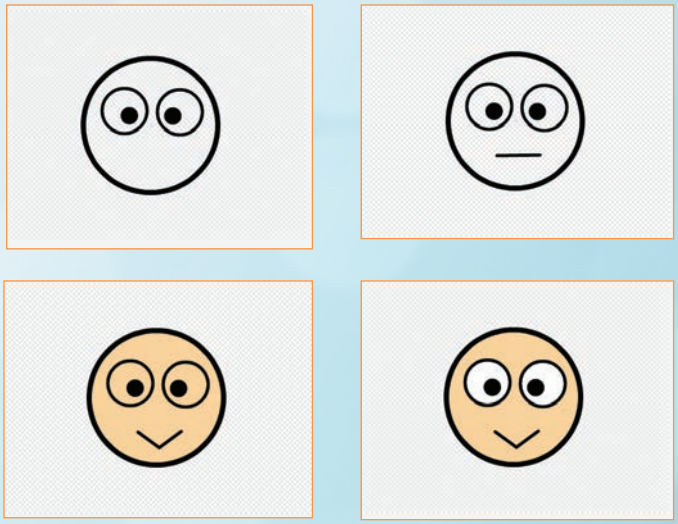




Çizim Araçları ve Animasyon

Bu Aktivitede Neler Öğreneceğiz?

- Noktasal ve çizgisel çizimler arasındaki farkı,
- Scratch üzerindeki çizim editör araçlarını,
- Kendi kılığını oluşturmayı,
- Animasyon oluşturmayı.



BİLGİ

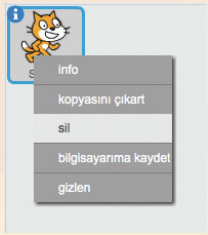
Scratch içine, internetten indirebileceğimiz **svg** uzantılı resimleri aktararak değişiklik yapabiliriz.

EK GÖREV

- Kendi emojilerimizi oluşturalım.
- Scratch kütüphanesinde var olan karakterler üzerinde değişiklik yapalım.
- Kendi karakterlerimizi ve kılıklarımızı oluşturalım.

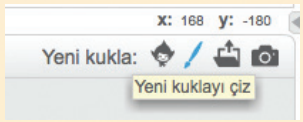
Bu bölümde Scratch içerisinde bulunan çizim araçlarını tanımaya çalışacağız. Araçları tanıdıktan sonra birlikte birkaç tane çizim yapacağız.

Scratch üzerinde iki tane çizim editörü var: Bunlar Bitmap (Noktasal) ve Çizgisel (Vektörel) çizim editörleridir. Çizim editörlerine ulaşmak için öncelikle kedimizi silelim. (Şekil-1)



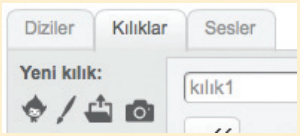
Şekil 1

Ardından "yeni bir kukla çiz" komutuyla kendimize boş bir kukla oluşturalım. (Şekil-2)



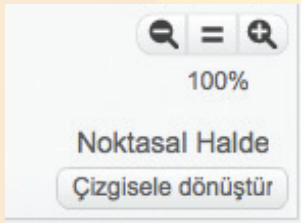
Şekil 2

Orta alanda bulunan Diziler, Kılıklar ve Sesler sekmelerinde **Kılıklar** sekmesinin seçilmiş olduğundan emin olalım. (Şekil-3)



Şekil 3

Sağ altta bulunan iki çizim moduna dikkat edelim. (Şekil-4)



Şekil 4

Önce Noktasal ve Çizgisel çizimler arasındaki farkı açıklayalım. Bitmap (noktasal) grafikler piksellerden oluşur. Pikseli ise bilgisayar ekranında görebileceğiniz en küçük nokta olarak tanımlayabiliriz. Vektör (çizgisel) grafikler ise nesnelerden oluşur. Bunları biraz daha derinlemesine anlatmaya çalışalım.

Bitmap - Vektörel

Dijital kameramızla ya da cep telefonumuzla bir fotoğraf çektiğimizde aslında noktasal (BitMap) bir grafik oluşturmuş oluyoruz.

Noktasal bir grafik adına piksel dediğimiz küçücük parçalardan oluşur ve bu piksellerin birçok rengi vardır. Bu piksellerin her birini tek tek değiştirme imkânımız vardır.

Bilgisayar bu pikselleri tek tek kaydettiğinden genellikle bu dosyaların büyüklüğü de biraz fazladır. Ve bu grafikleri büyüttüğümüzde kalitesinin düştüğünü görebiliriz. **https://tinyurl.com/bitmap-vector-1** adresindeki resmi yakınlaştırıp bunu görebiliriz.

Peki başka bir alternatif mümkün mü? Yine matematik yardımımıza koşuyor. Bu işin matematiğini anlatmak bu kitabın çok dışında olsa da bu konuda merak edip araştırmak ve okumak isteyebilirsiniz. Belki **Matematik Köyü**'ne gider sorularınızı sorar belki erken yaştan **Matematik Dünyası** alıp okumaya başlayabilirsiniz. Biz anlamakta zorlanıyoruz diye siz de zorlanmak zorunda değilsiniz.

Neyse konumuza dönelim.

Çizgisel (Vektörel) Grafik Nedir?

Çizgisel (vektörel) grafikler nesnelerden oluşur ve her bir nesnenin rengini, yerini, şeklini değiştirebiliriz. Scratch üzerindeki editörde çalışırken bunu çok daha iyi fark edeceksiniz.

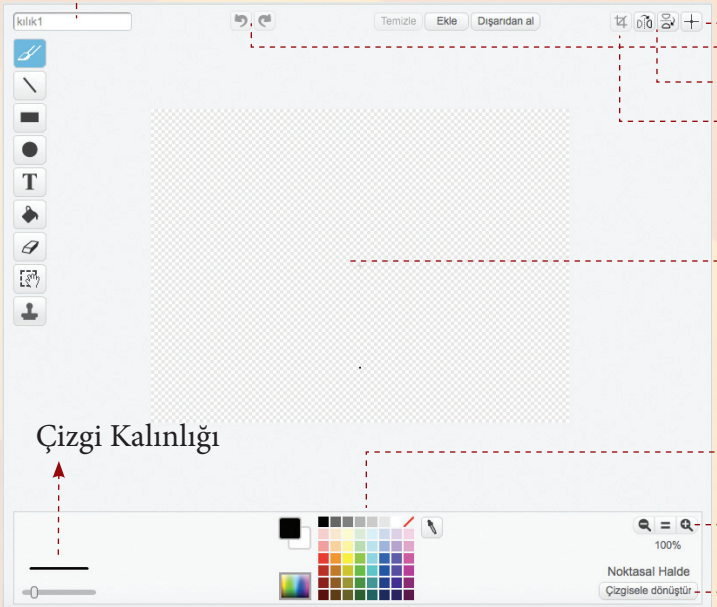
Vektörel grafikler büyük olsalar bile bilgisayar hafızasında çok az yer tutacaklardır. İstediğimiz kadar büyütüp küçültsek bile grafik üzerinde herhangi bir kalite kaybı olmayacaktır.

https://tinyurl.com/bitmap-vector-2

adresinde bunu deneyebiliriz.

Şimdi gelelim Scratch üzerinde grafik editor araçlarını tanımaya.

Scratch Çizim Araçlarını Tanıyalım



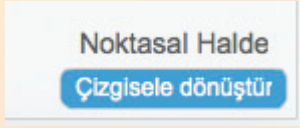
- Karakter ismi
- Kılık merkezi
- İleri geri al
- Şekli çevir
- Resmi kırp
- Çizim alanı
- Çizgi Kalınlığı
- Renk paleti
- Yakınlaştırma
- Editör seçimi

- Fırça
- Çizgi
- Dikdörtgen
- Elips
- Metin
- Renkle doldur
- Sil
- Seç
- İz bırak

- Seç
- Şekli değiştir
- Kalem
- Çizgi
- Dikdörtgen
- Elips
- Metin
- Renkle doldur
- İz bırak

Bu bölümde yapacağımız çizimleri Çizgisel (Vektörel) çizim editörü ile yapacağız. Birlikte bir yüz çizelim.

Kedimizi silmiş ve yeni kuklamızı oluşturmuş olarak çizimimize başlayalım. Öncelikle, Çizgisel Mod'a geçtiğimizden emin olalım. Bunun için "**Çizgisele Dönüştür**" düğmesine tıklamamız yeterlidir. (Şekil-8)



Şekil 8

Hâlihazırda zaten çizgisel moddaysak çizim araçlarının sağ tarafta olduğunu göreceğiz. (Şekil-9)

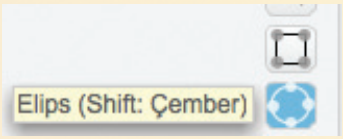
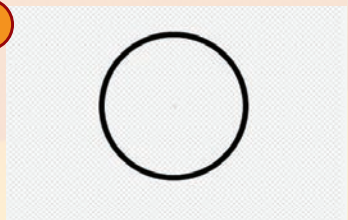
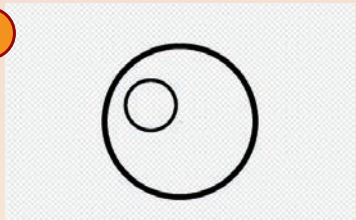
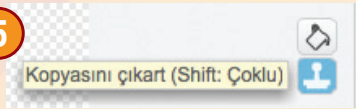
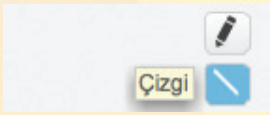
Şekil 9

BİLGİ

Scratch programlama dilinden türemiş birçok programlama dili mevcuttur. Bunlardan en geniş kullanımı olanlardan biri Berkeley Üniversitesi tarafından geliştirilmiş olan **Snap** dilidir. HTML5 tabanlı olan bu dile ulaşmak için **snap.berkeley.edu** adresini ziyaret edebilirsiniz.

Çizim Yapalım

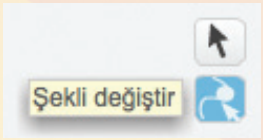
Başlamadan önce, önceki sayfada tanıttığımız bu araçların tümünü bir süre denemenizi şiddetle tavsiye ediyoruz. Çizim araçlarından Elips'i alıp başlayalım. Aşağıdaki adımları takip ederek ilk çizimimizi yapalım

1  Elips Çizim aracını alalım.	2 Editörün sağ altında bulunan bölümden şeklin içini boş olarak seçelim ve çizgi kalınlığını ayarlayalım.	3  Shift tuşuna basarak istediğimiz büyüklükte bir çember çizelim.
4  Elips aracı seçiliyken gözler için bir çember daha çizelim. İstersek bu çemberin çizgi kalınlığını değiştirebiliriz.	5  Araç çubuğunun alt bölümünde bulunan Kopyasını Çıkart düğmesini seçip çizdiğimiz gözün üzerine tıklayalım ve gözün bir kopyasını çıkartalım.	6 Oluşturduğumuz kopyayı dilediğimiz bir yere taşıyalım.
7 Çember aracı seçiliyken bu sefer içini dolu seçerek göz bebeğini oluşturalım.	8 Göz bebeğini gözün içinde istediğimiz bölgeye taşıyalım.	9  Araç çubuğundan Çizgi aracını seçelim.

10

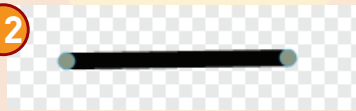
İstediğimiz büyüklükte bir ağız çizelim.

11



Bu çizgiyi değiştirmek için araç çubuğunda Şekli Değiştir düğmesine tıklayabiliriz.

12



Şekli Değiştir aracı seçiliyken çizginin üzerine tıklayalım. Çizgi üzerinde, 2 noktanın seçili olduğunu göreceğiz. Bu noktalardan çizgiyi uzatıp kısaltabiliriz.

13

Çizgi üzerinde başka değişiklikler yapmak için çizgiye eklemek istediğiniz noktanın yerini seçelim. Örnekte orta yerde bir nokta var.

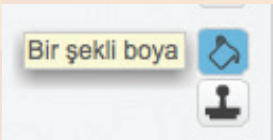
14

Bu noktayı sürükleyerek farklı mimikler elde edebiliriz.

15

Yaptığımız her adımda, bir önceki adımı kaybetmemek için kılığa sağ tıklayarak kopyalar çıkarabiliriz.

16



Araç çubuğundan "**Bir Şekli Boya**" düğmesini tıklayalım.

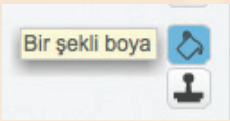
17

İstediğimiz bir rengi seçelim.

18

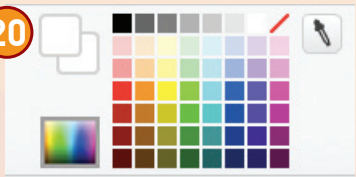
Boyamak istediğimiz bölümün üzerine gelip tıklayalım.

19



Bir Şekli Boya aracının seçili olup olmadığını kontrol edelim.

20



Gözler için beyaz rengi seçelim.

21

Son olarak gözleri boyayalım.

Bu ve benzeri birçok çizim yapabilir ya da var olan çizimlerin üzerinde değişiklik yaparak kendi karakterlerimizi oluşturabiliriz.

Elbette bu araçlarla yapabileceklerimiz çok fazla. Bundan sonraki bölümlerde kendinize yeni karakterler çizebilir ya da kodlama yaparken kullandığımız karakterleri değiştirebilirsin. Başka nasıl çalışmalar yapıldığını anlamak isterseniz **scratch.mit.edu** adresinde gezinmenizi tavsiye ederiz. Biz sizin için birkaç çizim projesi seçtik umarım beğenirsiniz;

https://scratch.mit.edu/projects/83031280/

https://scratch.mit.edu/projects/36520534/

https://scratch.mit.edu/projects/60143994/

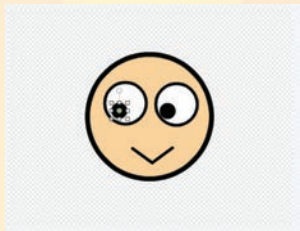
Peki Scratch kütüphanesinde bulunan, internet üzerinde bulduğumuz ya da farklı bir program ile çizdiğimiz bir resmi Scratch ile değiştirmemiz mümkün mü? Tüm resim formatlarını Scratch içerisine aktarabiliriz. Bu noktada resim üzerinde nasıl değişiklik yapabileceğimiz resmin uzantısına bağlı. Eğer .jpg, gif gibi uzantılarla bir resim indirdiysek bu resimlerle noktasal editör ile değişiklik yapabiliriz. Uzantısı .svg olan bir resim indirirsek çizgisel olarak istediğimiz değişikliği yapabiliriz.

Birden Çok Kılık Oluşturmak

Şimdi dilerseniz çizdiğimiz yüzü birçok şekle sokup küçük bir animasyon yapalım. En son çizdiğimiz yüzün üzerinde birkaç değişiklik yapalım. İlk olarak, sadece gözleri oynatalım. Bunun için daha önce çizmiş olduğumuz kılığın üzerine gelip, farenin sağ düğmesi ile bir kopyasını çıkartalım. (Şekil-30)

Şekil 30

Araçlar sekmesinden **Seç** aracını işaretledikten sonra, gözleri istediğimiz noktaya kaydıralım. (Şekil-31)

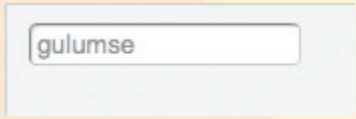


Şekil 31

Animasyonunuz için kaç tane istersek o kadar kopya çıkarıp gözü hareket ettirebiliriz. (Şekil-32)

Şekil 32

Dilersek bu kılıkların isimlerini istediğimiz şekilde değiştirebiliriz. Özellikle karmaşık projelerde bu işimizi çok kolaylaştıracaktır. (Şekil-33)



Şekil 33

Şimdi, "**Diziler**" sekmesini işaretleyip kod yazmaya başlayabiliriz. Boşluk tuşuna bastıkça, kuklamızın bir sonraki kılığa geçmesini sağlayabiliriz. Böylece, gözleri fırıl fırıl dönen bir karakterimiz olacak. Bunun için, görünüm sekmesinin altında bulunan "**Sonraki Kılık**" kod bloğunu kullanacağız. (Şekil-34)

Şekil 34

Bu kod bloğunu çalıştırdığımızda, gözlerin çok hızlı hareket ettiğini göreceğiz. Scratch saniyede 30 kare oynatabilir. Bu durumda iki yolumuz var; ya kılık sayısını artıracağız ya da araya bekleme süresi koyacağız. Biz, daha kolay bir şekilde aynı etkiyi uyandıracağımız ikinci yolu seçelim ve sadece bir kod bloğuyla animasyon süresini istediğimiz gibi ayarlayalım. (Şekil-35)

Şekil 35

Süreyi değiştirerek, istediğimiz şekilde animasyonu hızlandırabilir ya da yavaşlatabiliriz. (Şekil-36)

Şekil 36

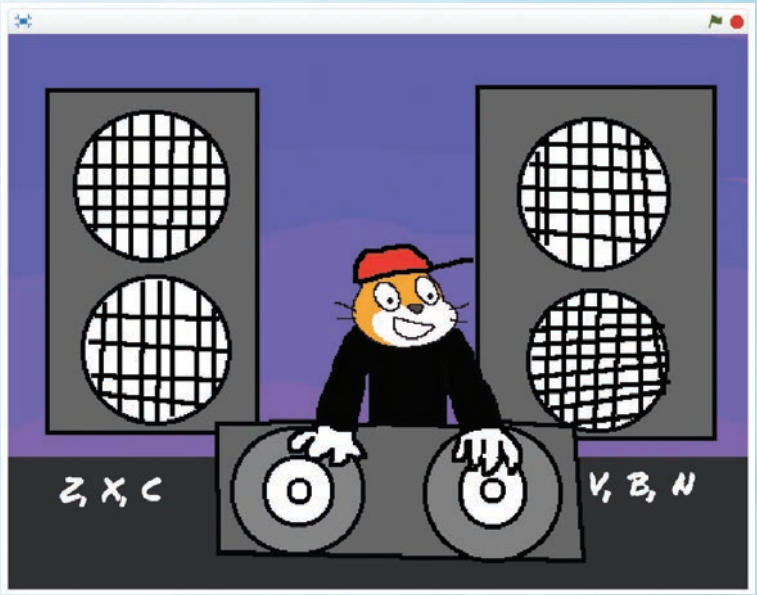
Scratch içinde ondalıklı bir sayı girmek istediğinizde bunu nokta ile yazmalısınız, virgül ile değil. Yani ½ yazmak için 0,5 değil 0.5 yazmalısınız.

Tebrikler başardınız!

DJ Scratch

Bu Aktivitede Neler Öğreneceğiz?

- Arka plan belirlemeyi,
- Scratch içinden ses dosyası oynatmayı,
- Sesi çal ile sesi bitene kadar çal arasındaki farkı.



BİLGİ

Dışarıdan aldığımız bir mp3 dosyasını scratch içine aktarabiliriz.

EK GÖREV

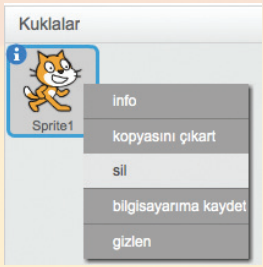
- Ses dosyalarını başka ses dosyaları ile değiştirelim.
- Daha fazla tuş ekleyelim.
- Kendi ses efektlerimizi oluşturalım.

Sahnemizi Düzenleyelim

Bu uygulamada, Scratch üzerinde ilk müzik uygulamamızı yapacağız. Scratch kedimiz bir DJ olacak ve klavyeden basılan tuşlara göre farklı sesler ve efektler çıkaracak.

Bu uygulama içinde kullanacağımız karakterleri ve arka planı **www.ogreniyorum.xyz** adresinde bulabilirsiniz.

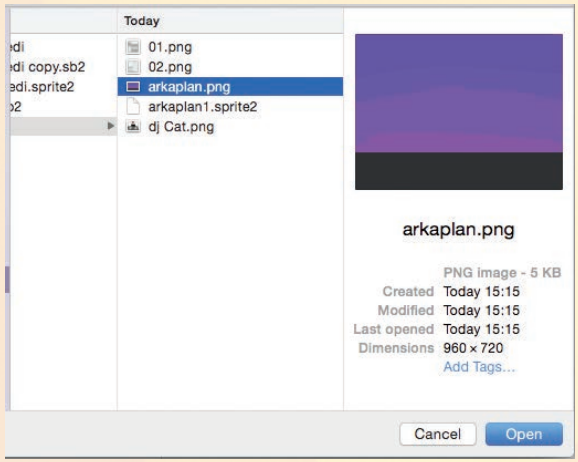
Uygulamamıza başlamadan önce, ekranda görünen kedimizi silelim. Bunun için kedimizin üzerine gelip sağ fare tuşu ile çıkan menüden sil komutunu seçelim. (Şekil-1)



Şekil 1

Scratch DJ uygulamasına arka plan ekleyelim.
Ekteki **arkaplan.png** dosyasını içeri aktaralım. (Şekil-2, 3)

Şekil 2



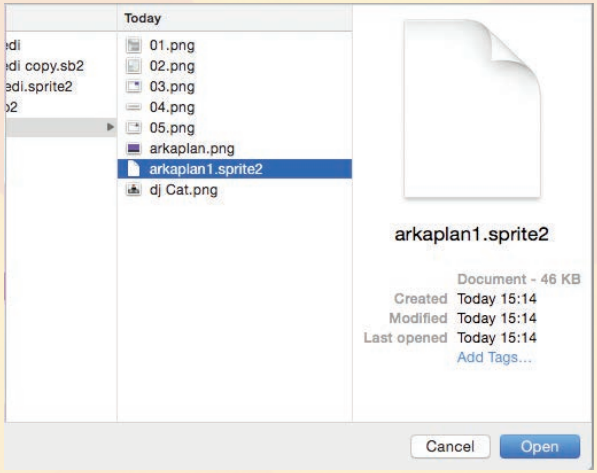
Şekil 3

Scratch kedimizin arkasında duracak olan hoparlörleri karakter olarak içeri aktaralım. Bunun için, içeri aktar komutunu kullanalım. (Şekil-4)

Şekil 4

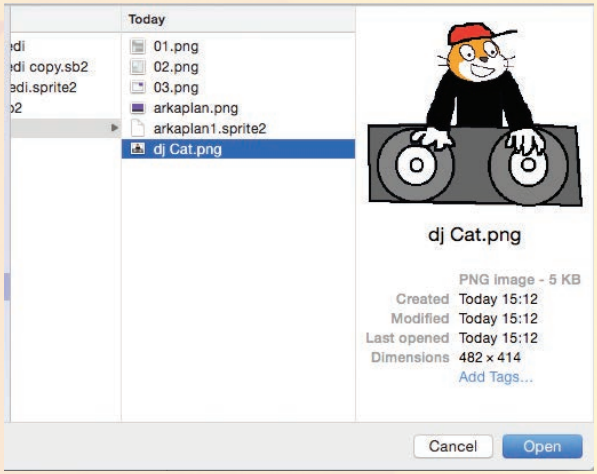
Eğer internet bağlantınız yoksa, siz de kendi sahnenizi oluşturabilirsiniz.

Ardından web sitesinden indirdiğimiz **arkaplan1.sprite2** dosyasını seçelim. (Şekil-5)



Şekil 5

Artık DJ'imizin sahneye çıkma vakti geldi. Son olarak karakterimizi -yani kedimizi- sahneye ekleyelim. (Şekil-6)



Şekil 6

BİLGİ

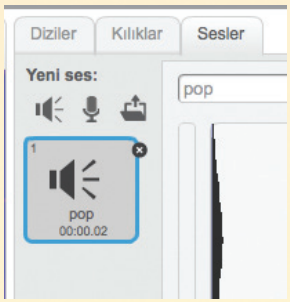
İnternet üzerinde **freesound.org** gibi sitelerden kişisel kullanımınız için ses dosyaları indirebilirsiniz. Bu ses dosyalarının formatlarını değiştirmek isterseniz **media.io** sitesini kullanabilirsiniz.

Şu an sahnemizin görünümü şu şekilde olmalıdır: (Şekil-7)

Şekil 7

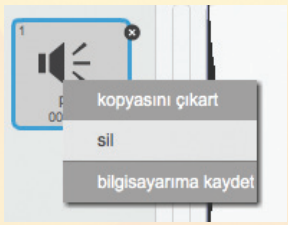
Ses Dosyası Ekleyelim

Kodlamaya geçmeden önce, DJ'imizin kullanacağı ses dosyalarını içeri aktarmalıyız. Bu bölüm için, indirdiğimiz dosyaların arasından ses klasörünü bulalım. Klasörün içerisinde 7 tane dosya göreceğiz. Bu dosyaları Scratch uygulamamızın içine aktaralım. Bunun için, orta bölümde bulunan ses sekmesini tıklayalım. (Şekil-8)



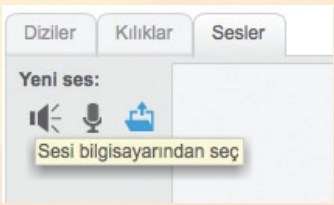
Şekil 8

Var olan "**pop**" adlı ses dosyasını silelim. Bunun için ses dosyasının üstüne sağ tıklayınca çıkan menüden "**sil**" komutunu seçelim. (Şekil-9)



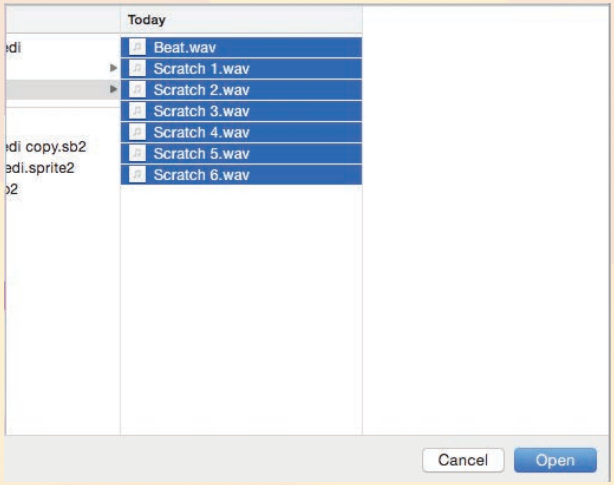
Şekil 9

Şimdi içeriye ses dosyalarını aktaralım. "**Sesi bilgisayarımdan seç**" düğmesine tıklayalım. (Şekil-10)



Şekil 10

Klasörümüze girip tüm dosyaları seçelim. Dosyaları tek tek içeri aktarabileceğimiz gibi, toplu olarak da içeri aktarabiliriz. (Şekil-11)



Şekil 11

Artık orta bölümde bulunan Diziler sekmesine tıklayarak kodlama yapmaya başlayabiliriz. (Şekil-12)

Şekil 12

Uygulama başladığında arka planda sürekli bir beat (ritim) çalmasını istiyoruz. (Şekil-13)

Şekil 13

Şimdi, klavyedeki farklı harflere bastıkça birbirinden değişik ses efektleri çalmayı deneyeceğiz. Bu sırada DJ'imiz de kılık değiştirecek. Tüm kod bloklarının öncelikle bir ses dosyasını çalmasını, bu sırada da önce kostum2'ye geçip kısa bir süre (0.2 saniye) bekledikten sonra kostum1 kılığına geçmesini sağlayacağız. (Şekil-14, 15)

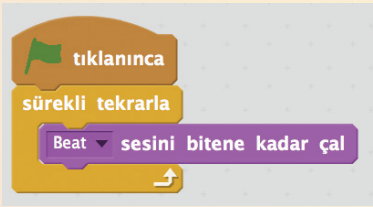
Artık uygulamanızın tadını çıkarma zamanı. İyi eğlenceler.

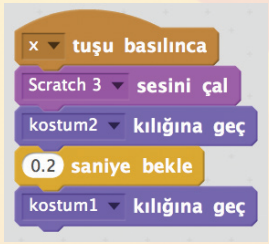
Şekil 14

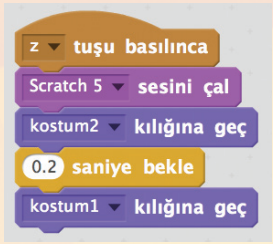
Şekil 15

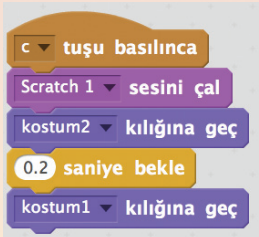
Scratch, MIT Medya Lab'ında yer alan Lifelong Kindergarten grubunun bir projesidir. Ve tamamen ücretsizdir.

BU BÖLÜMDE KULLANILAN KOD BLOKLARI



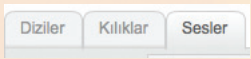






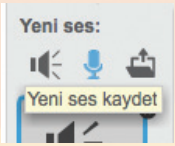
ÇALIŞMA SAYFASI

1. Kullandığımız bilgisayara takılı bir mikrofon varsa ses kaydı yapıp bunlar üzerinde değişiklik yapabiliriz. Scratch programının orta bölümünde bulunan Sesler sekmesini açalım. (Şekil-1)



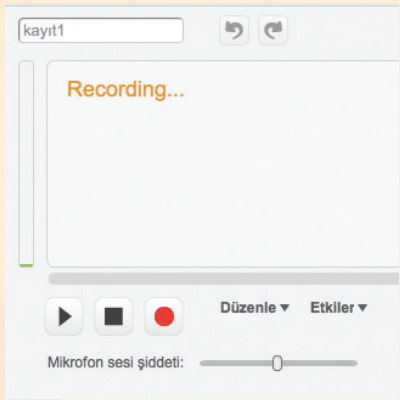
Şekil 1

2. Yeni ses dosyası açmak için mikrofon ikonuna tıklayalım. (Şekil-2)



Şekil 2

3. Yeni ses kaydı yapmak için Kaydet düğmesine tıklayalım. (Şekil-3)

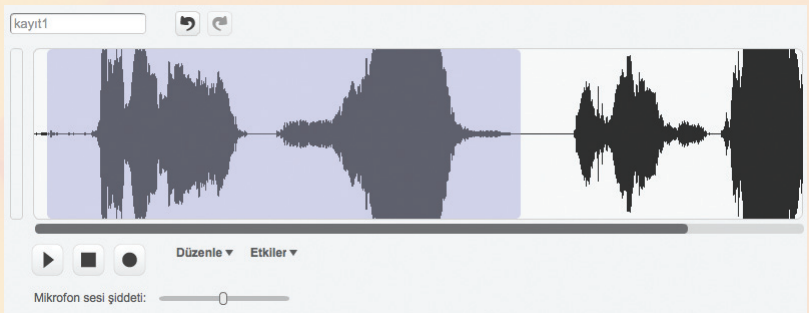


Şekil 3

4. Mikrofondan sesimizi kaydedelim. Ses kaydını tamamlamak için orta bölümde bulunan durdur düğmesine tıklayalım. (Şekil-4)

Şekil 4

5. Ardından bu sesin üzerinde oynamalar yapacağız. Sesin üzerinde değişiklik yapmak istediğimiz alanı seçelim. (Şekil-5)

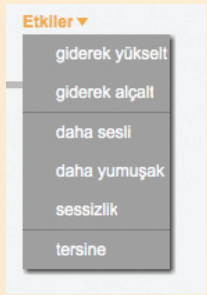


Şekil 5

6. Bu alanı seçtikten sonra sesin üzerinde istediğimiz değişiklikleri yapabiliriz. Düzenle bölümünden; kopyalamak, kesmek gibi işlemleri yapabiliriz. (Şekil-6)

Şekil 6

7. Etkiler bölümünden seçtiğimiz ses efektini uygulayabiliriz. (Şekil-7)



Şekil 7

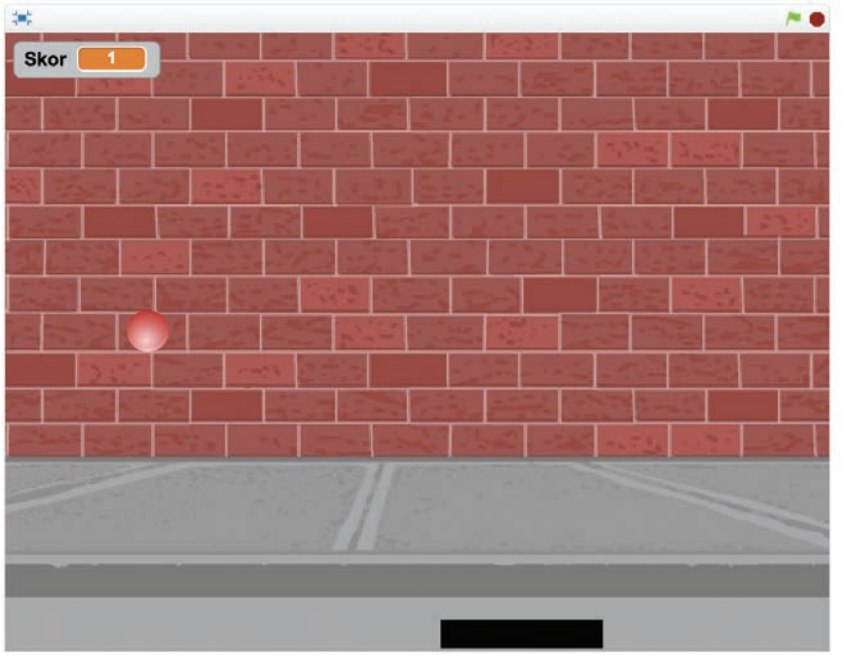
EK GÖREV

Var olan sesler üzerinde ya da kendi kaydettiğimiz ses üzerinde çeşitli denemeler yapalım.

Pong

Bu Aktivitede Neler Öğreneceğiz?

- Klavye ya da fare ile karakter kontrolünü nasıl yapacağımızı,
- PONG oyununu Scratch kullanarak nasıl yapabileceğimizi,
- Algılama sekmesini kullanmayı.



BİLGİ

Pong oyununun değişik versiyonlarını **http://www.ponggame.org/** adresinde bulabilirsiniz.

EK GÖREV

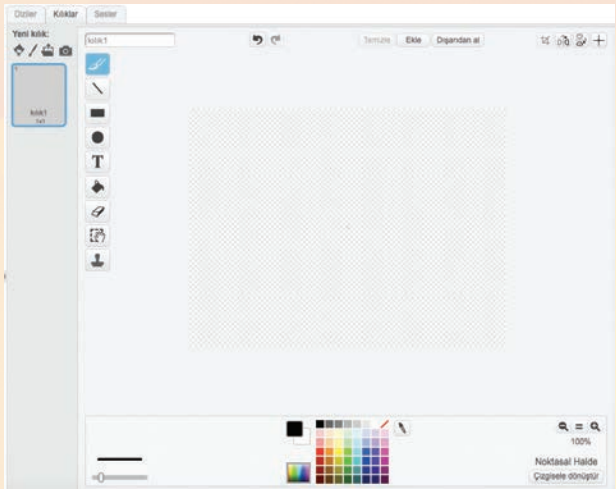
- Oyunu iki kişilik yapalım.
- Oyunu dikey olarak oynanabilir hale getirelim.
- Belirli puanları geçtikçe oyunun hızını artıralım.

Raketimizi Oluşturalım

Bu oyunda iki nesnemiz var: biri raket diğeri top. Grafik düzenleyiciyi kullanarak topumuzu ve raketimizi oluşturalım. Yeni kukla çiz komutuna tıklayarak çizim ekranımızı açalım. (Şekil-1)

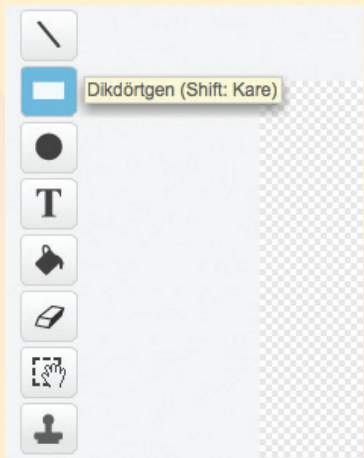
Şekil 1

Açılan ekranda birçok temel çizim aracı bulunmaktadır. (Şekil-2)



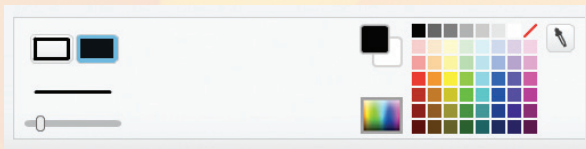
Şekil 2

Raketimizi oluşturmak için dikdörtgen aracını seçelim. (Şekil-3)



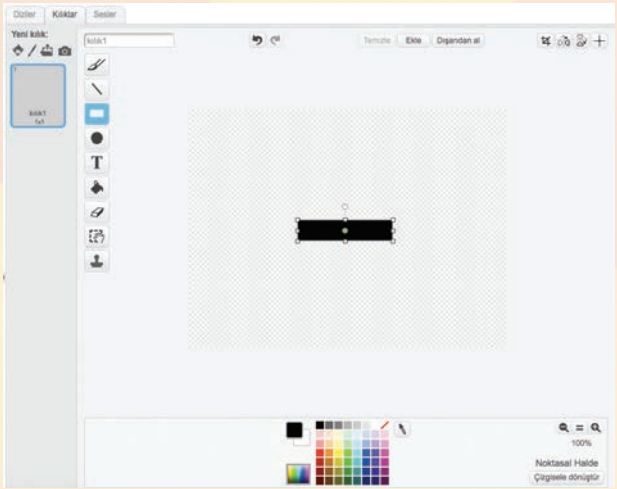
Şekil 3

İçini dolu çizecek biçimde seçimimizi yapalım. (Şekil-4)



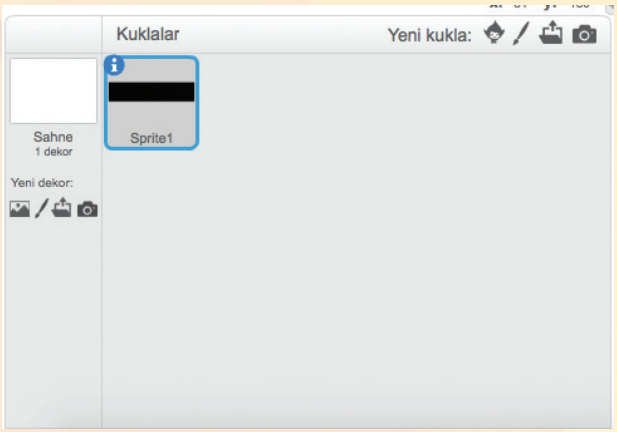
Şekil 4

Raketimizi istediğimiz boyutta çizelim. (Şekil-5)



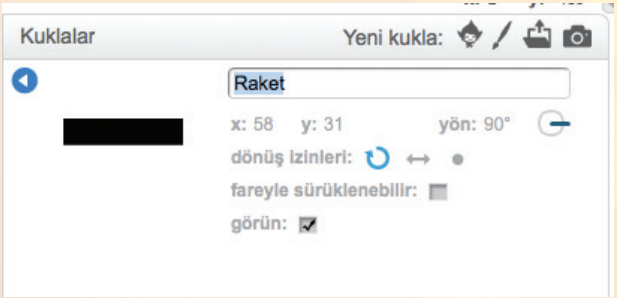
Şekil 5

Raketimize ismini verelim. Rakete ismini vermek için karakterlerin bulunduğu bölümde mavi "i" harfinin üzerine tıklıyoruz. (Şekil-6)



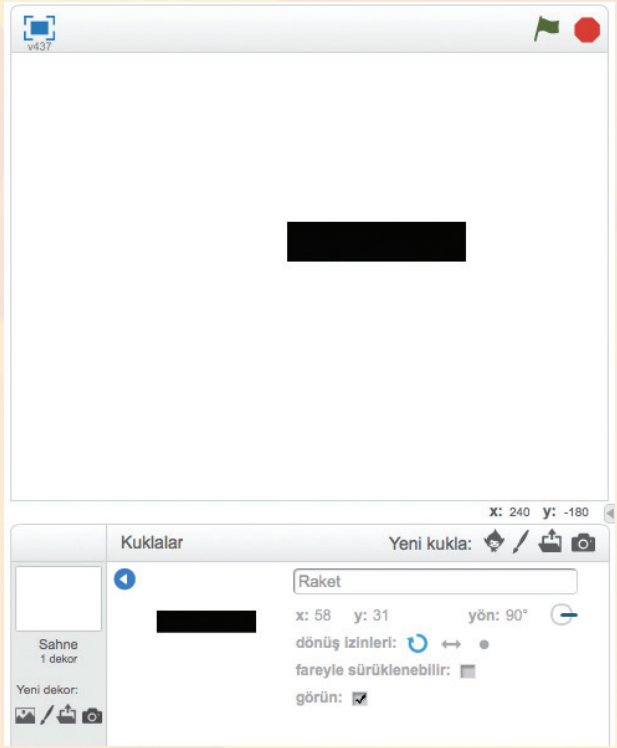
Şekil 6

"İ" tuşuna bastığımızda bir arayüz görünüyor. Bu alandan raketimizin yönünü değiştirebilir, ekranda görünüp görünmemesini belirleyebiliriz (Bu ekranda yapabildiğimiz her şeyi kod blokları ile de yapmak mümkündür). Dönüş izinlerini ayarlayabileceğimiz gibi, oyun içerisinde oyuncu tarafından faresiyle bir yerden başka bir yere taşınıp taşınamayacağının izinlerini bulabiliriz. (Şekil-7)



Şekil 7

Raketimizi oluşturduğumuza göre şimdi hareket etmesini sağlayalım. Raketimizin sadece sağa sola hareket etmesi gerekiyor. Bunu yön tuşlarıyla yapabileceğimiz gibi klavyeden "a" ve "d" tuşlarıyla da gerçekleştirebiliriz ya da raketimizin faremizi takip etmesini sağlayabiliriz. Biz klavyenin sağ-sol tuşlarıyla hareket edecek şekilde raketimizi programlayalım. Bunlardan önce, raketimizi oyunun başlangıcında nerede olmasını istiyorsak oraya sürükleyelim. (Şekil-8)



Şekil 8

Her oyun başladığında (yani yeşil bayrak tıklanınca) raketimizin olmasını istediğimiz yere gitmesini sağlayan kod bloğumuzu oluşturalım. (Şekil-9)

Şekil 9

Kontrol sekmesinin altındaki "**boşluk tuşu basılınca**" bloğunu sürükleyip bırakalım. Ortaya çıkacak menüden "sağ ok"u seçelim.
(Şekil-10)

Şekil 10

Sağ ok her basıldığında, raketimizin yatay eksende (x-ekseni) 7 birim sağa gitmesini (bu sayıyı değiştirebilirsiniz) istiyoruz. (Şekil-11)

Aynı kod bloğundan bir tane kopyalayıp "sol ok tuşu basılınca" -7 adım git bloğunu oluşturmalıyız. (Şekil-11)

Şekil 11

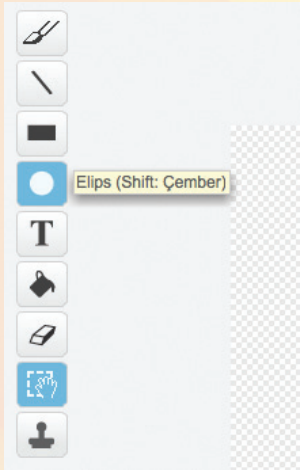
Klavyeden komut almanın başka yolları da var. Şekil-12'deki kod uygulamasını yapalım. Tabii ki Şekil-11'deki kodları silerek. Aradaki farkı gözlemleyelim.

İkinci yaptığımız kod bloğu çok daha güzel çalıştı. Bu durumu şöyle açıklayabiliriz: Birinci durumda kod blokları sadece basılma anında çalışırken, ikinci durumda tuşa basılı olduğu süre boyunca çalışan bir kod bloğumuz var.

Şekil 12

Topumuzu Oluşturalım

Raketimizi hareket ettirebiliyoruz fakat ortada henüz bir topumuz yok. Kendimize "top" isimli bir karakter oluşturalım. Yeni kuklayı çiz düğmesine tıklayıp çıkan menüden Elips aracını seçelim. Dikkat ederseniz, Elips aracının üzerine geldiğinizde "Shift: Çember" gibi bir açıklama göreceksiniz. (Şekil-13)

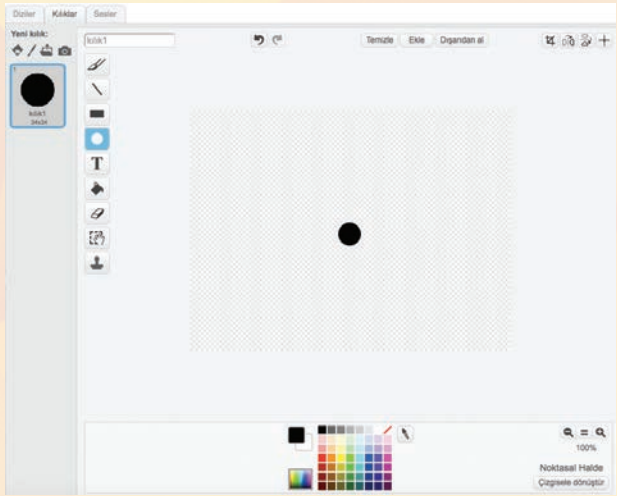


Şekil 13

Elips aracıyla çizim yapmaya çalıştığımızda tam bir çember ya da daire çizmek oldukça güç. Eğer Shift tuşuna basılı tutarak Elips aracını kullanırsak, çember ya da dairenin çizildiğini görürüz. Şimdi, topu istediğimiz büyüklükte, Shift tuşu yardımıyla çizelim. (Şekil-14)

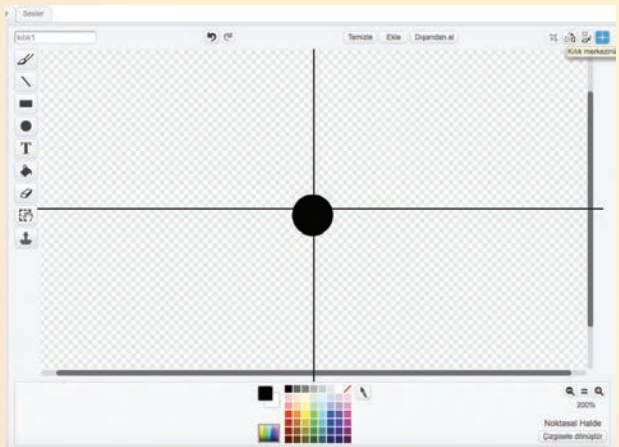
BİLGİ

Shift tuşu genellikle klavyenin sağ ve sol tarafında bulunur ve üzerinde yukarı ok sembolü vardır.



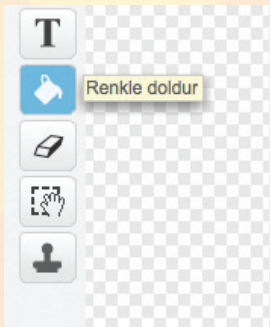
Şekil 14

Çizim yaparken nesnenin orta noktası Scratch çizim alanının da orta noktası olmalı, aksi halde nesnenin tam yerini tayin edemeyiz. Peki, orta noktaya göre çizim yapmadığımızda ya da sonradan şekil üzerinde oynama yaptığımızdan dolayı merkez bozulduğunda ne yapmalıyız? Bu durumda, çizim alanının sağ üst kısmında bulunan "kılık merkezi belirle" düğmesine tıklayıp merkezi yeniden belirleyebiliriz. (Şekil-15)



Şekil 15

Topumuza biraz üç boyut efekti katarsak çok daha güzel görünecek. Bunun için "**renkle doldur**" düğmesine tıklayalım. (Şekil-16)

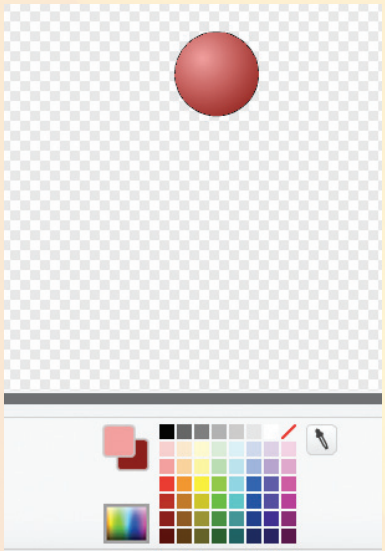


Şekil 16

Alt tarafta çıkacak olan seçeneklerden "gradient" efektini merkezden çevreye azalacak şekilde seçelim. (Şekil-17)

Şekil 17

Renk olarak topun olmasını istediğimiz rengin açık ve koyu tonlarını seçip, açık ton üstte olacak şekilde seçim yapalım. Topun sol üst ya da sağ üstüne tıkladığımızda topumuzda 3B efektini gözlemleyebiliriz.
(Şekil-18)

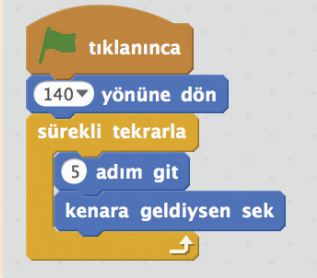


Şekil 18

Gelelim topumuzun hareketini sağlayacak komutlara. Başlangıç değeri olarak topumuzun belirli bir derece ile bir yöne dönmesini ve daha sonra sürekli olarak hareket etmesini istiyoruz. (Şekil-19)

Şekil 19

Şekil-19'daki kodu çalıştırdığımızda topumuzun köşeye kadar gidip orada sıkıştığını göreceğiz. Topumuzun kenara geldiğinde ne yapacağını ya da yeşil bayrağa tıkladığımızda topun nereden başlayacağını belirtmedik. Kodumuzu düzeltip tekrar bakalım. Şekil-20'de topumuza (0,100) gibi bir başlangıç değeri verdik, topumuzun hızını 5 adıma düşürdük ve kenara geldiğinde sekmesini istedik. Yeşil bayrağa tıkladığımızda topumuz sürekli hareket ediyor fakat rakete değdiğinde herhangi bir değişiklik olmuyor ya da oyun hiçbir zaman bitmiyor. (Şekil-20)



Şekil 20

Topun Sekmesini Sağlayalım

Bir önceki kod bloğumuza top rakete değdiğinde sekmesini sağlayacak kod bloğunu ekleyelim. Eğer top rakete değerse, bütünler açısı kadar (180°–yön) dönmesini sağlayalım.

Yaptığımız ek ile eğer top rakete değerse "180° – yönü" kadar dönecek, yani sekmiş olacak. Sürekli olarak rakete değip değmediğini kontrol ediyor ve eğer değdiyse "180 – yönü" kadar dönmesini istiyoruz. Programımızı çalıştıralım. Şu an raketle karşılama yapabiliyor fakat raket ile topu karşılayamasak da topun alt zemine çarpıp sektiğini görüyoruz. (Şekil-21)

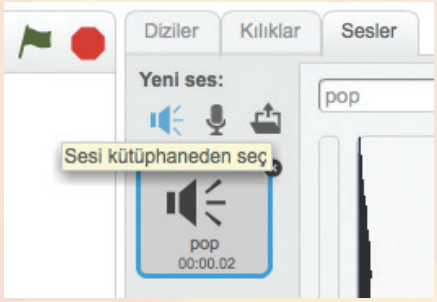
Şekil 21

Top, raketin altında bir yere düştüğünde oyunun durmasını sağlayalım. Sürekli olarak eğer topun y konumu **–160° değerinden küçük mü**, diye kontrol edelim. Eğer y konumu –160° değerinden küçükse, oyunun durmasını sağlayalım.

Top raket ile karşılanamadığında oyun bitmeli. Şimdilik bir canımız olduğunu düşünelim. Topun daha alt seviyeye düşmesi demek y ekseninde daha düşük bir seviyede olması demektir. Scratch ekranı y ekseninde –180° ile +180° arasında değişiyor. Biz –160°'den aşağı bir seviyede mi diye kontrol edersek topun raketten daha alt bir seviyeye düşüp düşmediğini kontrol etmiş oluruz. (Şekil-22)

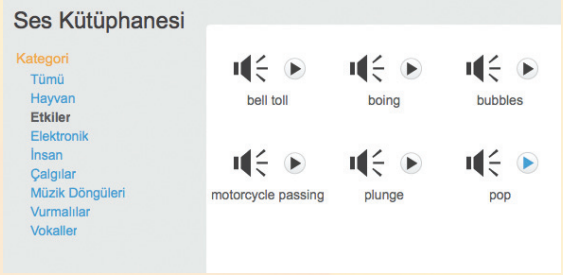
Şekil 22

Oyunumuz bu haliyle çalışır durumda, yalnız bir iki küçük şey ekleyerek biraz daha güzel hale getirebiliriz. Öncelikle küçük bir ses efekti ekleyebilir ardından bir oyuncunun kaç puan aldığını belirleyebiliriz. Ses efekti eklemek için öncelikle dışarıdan ses yüklemeliyiz. Bu ses bilgisayarımızdan herhangi bir ses olabileceği gibi Scratch'in kendi kütüphanesinden de bir ses seçebiliriz. (Şekil-23)



Şekil 23

Scratch kütüphanesinden "Etkiler" sekmesinin altında **pop** sesini seçelim. (Şekil-24)

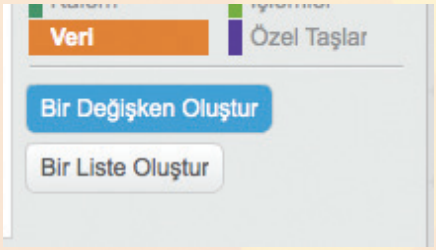


Şekil 24

Pop sesini programımıza ekledikten sonra yapmamız gereken tek şey, top rakete değdikçe bu sesi oynatmasını istemek. Bunu SES sekmesinin altında "**... sesini çal**" bloğu ile gerçekleştirebiliriz. (Şekil-25)

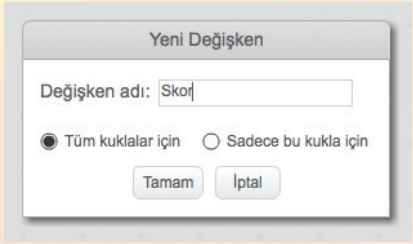
Şekil 25

Son olarak oyunumuza skor ekleyelim. Bunun için, Veri sekmesini tıklayalım. (Şekil-26)



Şekil 26

Açılan alt sekmeden "**bir değişken oluştur**" düğmesini tıklayıp Skor adlı değişkeni oluşturalım. (Şekil-27)

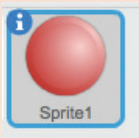


Şekil 27

Değişkenimizi oluşturduktan sonra değişkenimize başlangıç değeri olarak sıfırı atayalım. Şimdi top rakete her değdiğinde değişkenimizi 1 artırarak oyun sonunda topun rakete kaç kez değdiğini görebiliriz. (Şekil-28)

Şekil 28

BU BÖLÜMDE KULLANILAN KOD BLOKLARI



ÇALIŞMA SAYFASI

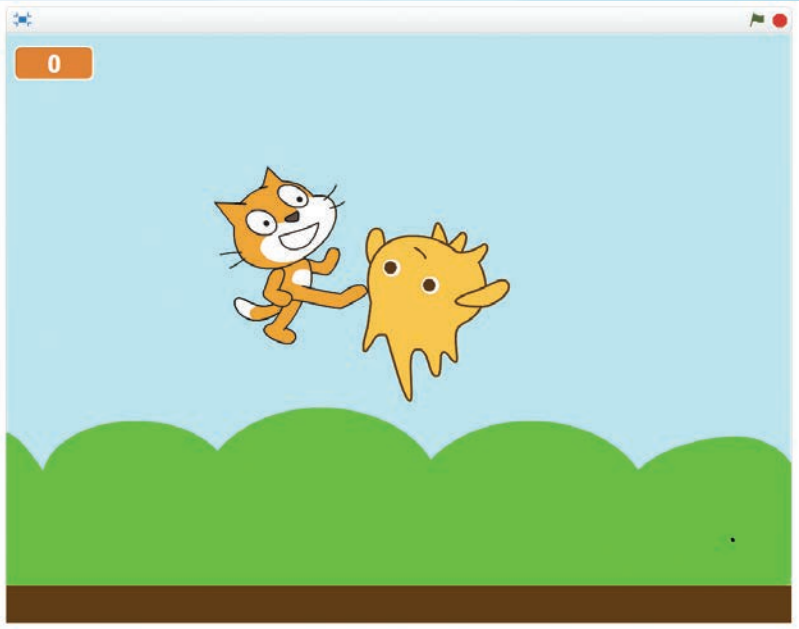
Tümler açı: Toplamlarının ölçüsü 90° olan açıları ifade eden geometrik terim.
Bütünler açı: Toplamlarının ölçüsü 180° olan açıları ifade eden geometrik terim.

Aşağıdaki şekillerde istenen açıları bulalım.

Kovalamaca

Bu Aktivitede Neler Öğreneceğiz?

- Bir karakterin fareyi takip etmesini sağlamayı,
- Algılama sekmesini,
- Zaman sayacı oluşturmayı,
- Değişken oluşturmayı.



BİLGİ

Scratch.mit.edu üzerinde Temmuz 2017 itibariyle 23,799,965 tane proje paylaşılmıştır.

EK GÖREV

- Gobo kediyi yakaladığında ses çıkarmasını sağlayalım.
- Gobo'nun daha hızlı hareket etmesini sağlayalım.
- Kedi ve Gobo karakterlerini değiştirelim.

Fare Okunu Takip Edelim

Bu bölümde **kedimiz** fare imlecini, **Gobo** karakterimiz ise kedimizi takip edecek. Belki yakalanana kadar geçen süreyi hesaplar, süre arttıkça daha hızlı hareket ederek oyunumuzu zorlaştırırız. Hadi başlayalım.

Öncelikle yeni bir Scratch projesi açalım. **Yeşil bayrak** tıklanır tıklanmaz kedimizin **sürekli olarak** fare okunu takip etmesini istiyoruz. (Şekil-1)

Şekil 1

Kedi karakterinin seçili olduğundan emin olalım. Bu kodu çalıştırdığımızda kedimizin hep fare okunun bulunduğu yerde olduğunu göreceğiz; fakat biz fare okunu takip etmesini istiyoruz. Bunu sağlamak için kedimizin fare okuna doğru dönmesini ve dönme yönünde ilerlemesini sağlayacağız.

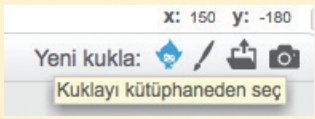
(Şekil-2, 3) Bir önceki kodla arasındaki farkı inceleyelim.

Şekil 2

Şekil 3

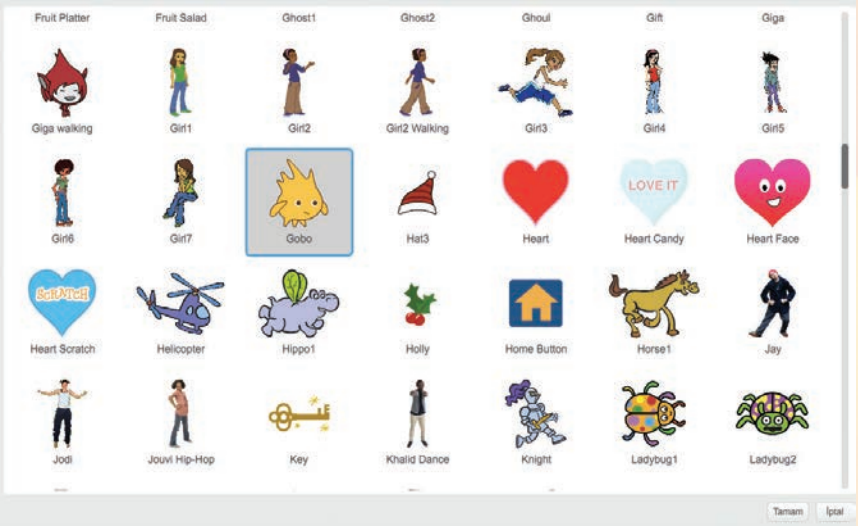
Eğer kedimizin hızını değiştirmek istiyorsak **10 adım git** komutundaki sayıyı değiştirmemiz yeterlidir.

Şimdi bir karakter daha ekleyelim ve onun da kedimize doğru hareket etmesini sağlayalım. Scratch kütüphanesinden yeni bir karakter eklemek için sol alt bölümde bulunan "**Kuklayı kütüphaneden seç**" düğmesine tıklayabiliriz.
(Şekil-4)



Şekil 4

Biz içeriye **Gobo** adındaki bir Scratch karakterini aktaracağız. (Şekil-5)



Şekil 5

Dilersek, içeriye başka bir karakter de aktarabiliriz. Gobo'nun yapacağı şey Scratch kedimize doğru dönüp ona doğru ilerlemek. Kedimiz için yazdığımız kodun bir benzerini yazacağız. (Şekil-6)

Şekil 6

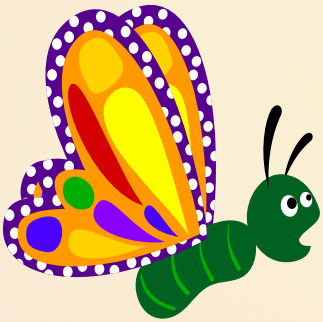
BİLGİ

Gobo'nun hızını kediden yavaş ayarladık ki kedimizin **Gobo**'dan kaçma fırsatı olsun. İki karakterin hızlarını değiştirerek istediğimiz gibi ayarlayabiliriz.

Oyunumuzu önce biraz oynayalım ve olası hataları tespit edelim. Oyunu biraz oynamaya çalıştığımızda, kedimizin farenin üzerine geldiğinde yönünü şaşırdığını ve bocaladığını görüyoruz. O halde, kedi fare okuna değince hareket etmemeli ya da diğer bir ifadeyle kedi fare okuna değmediği sürece hareket etmeli. (Şekil-7)

Şekil 7

Scratch içinde karakterlerin birbirine değip değmediğini anlamak için gerekli kod blokları **Algılama** sekmesinin altındadır. (Şekil-8)

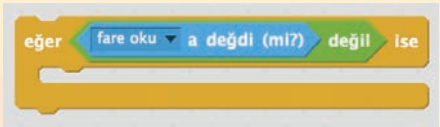


Şekil 8

Şekil-7'deki kodu bir defada yapamayanlar için adım adım resimleri: (Şekil 9, 10, 11)

Şekil 9

Şekil 10



Şekil 11

Şimdi uygulamamıza biraz daha oyun havası katmak için birkaç değişiklik yapalım.

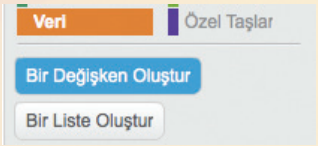
Kendi Zaman Sayacımızı Oluşturalım

Zaman sayacı oluşturmak için sürekli ilerleyen değişen bir yapıya ihtiyacımız var. Bunun cebirsel karşılığı **değişken**, Scratch üzerindeki karşılığı ise **Veri**'dir. Öncelikle **Sayaç** diye bir değişken oluşturmalı ve her saniye bunun bir artmasını sağlamalıyız.

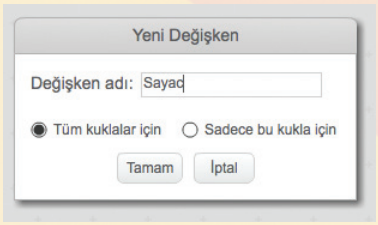
Değişken oluşturmak için, orta alanda bulunan veri sekmesine tıklayalım. (Şekil-12)

Şekil 12

Veri sekmesinin altında 2 ayrı komut düğmesi var: "**Bir değişken oluştur**" ve "**Bir liste oluştur**". Liste kullanmayı kitabımızın son bölümlerine doğru öğreneceğiz. Şimdilik ihtiyacımız olan "**Bir Değişken Oluştur**" düğmesine tıklayarak değişken oluşturmak. Oluşturduğumuz değişkenin adını **Sayac** olarak belirledik.
(Şekil-13, 14)



Şekil 13



Şekil 14

Değişkeni oluşturur oluşturmaz ortaya çıkan kod bloklarına dikkat ettiniz mi?
(Şekil-15)

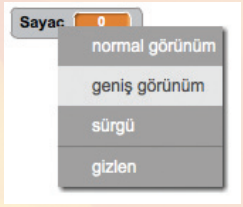
Şekil 15

Sayac değişkenini "0" değeriyle başlatıp her geçen saniyede 1 artıralım.
(Şekil-16)

Şekil 16

Ekranın sol üstünde bulunan Sayac değişkeninin görünümünü değiştirmek istersek, **Sayac** değişkeninin üzerine gelip farenin sağ tuşu ile tıklayalım. Açılan menüden **geniş görünüm'**ü seçelim.
(Şekil-17, 18, 19)

Şekil 17



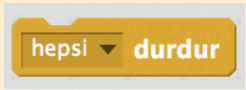
Şekil 18

Şekil 19

Bu sayaç, oyun başlayınca yani yeşil bayrağa tıklanınca başlayacak. (Şekil-20)

Şekil 20

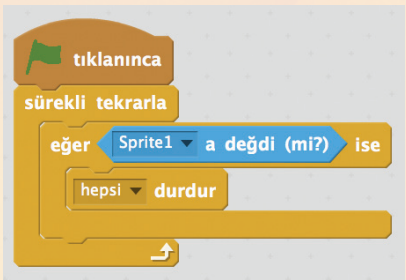
Oyunumuzu sonlandırmak ise Gobo'nun kedimizi yakalamasıyla olacak. Bunun için Gobo kediyi yakalayınca oyunu durduralım. Kuklaları durdurmak için **Kontrol** sekmesinin altında bulunan "**... durdur**" komutunu kullanacağız. (Şekil-21)



Şekil 21

Oyunu durdurmak için yazdığımız kod:

1. Oyun başladığında
2. Sürekli olarak kontrol et
3. Eğer Kediye değersen
4. Hepsini Durdur



Şekil 22

Şu an Gobo Karakterinin üzerindeyken kod yazıyoruz.

Oyunu tekrar başlattığınızda karakterler birbirine değiyorsa, oyun başlar başlamaz duracaktır. Fare imlecinizle karakterleri istediğiniz yere taşıyın.

BU BÖLÜMDE KULLANILAN KOD BLOKLARI

ÇALIŞMA SAYFASI

1. Gobo karakteri ile kedi kovalamaca oynamaktadır. Gobo karakterimiz kedi karakterinden kaçmaktadır. Aşağıdaki kod bloklarına göre 2 dakika sonunda yakalanan Gobo'nın elde ettiği puanı hesaplayalım.

2. Aşağıda verilen kod bloklarında bulunan boşlukları hikâyemize uygun olarak dolduralım.

   a. Kedimiz Gobo karakterinden kaçmaktadır. Başlangıçta 100 puanı bulunan kedimizin her saniye 5 puan kazanmasını istemekteyiz.

   b. Gobo karakterimiz kediyi kovalamaktadır. Başlangıçta 100 puanı olan Gobo'nun kediyi yakalayamadığı her 2 saniyede 6 puanı azalmaktadır.

   c. Gobo'nun 1 dakika boyunca kediyi yakalayamadığını bildiğimize göre, Gobo'nun 36 puanı kalması için kaç saniye daha yakalayamaması gerekmektedir?

İsmimizi Canlandıralım

Bu Aktivitede Neler Öğreneceğiz?

- Hayalet görünüm efektini kullanmayı,
- Balık gözü efektini kullanmayı,
- Hayalet efektini kullanmayı,
- Süzül kod bloğunu kullanmayı,
- Animasyon hazırlamayı.

BİLGİ

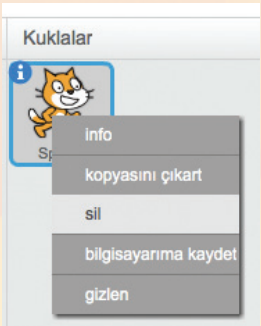
İnternet üzerinden bulabileceğiniz hareketli resimleri (animated gif) Scratch içerisine atabilirsiniz.

EK GÖREV

- Kendi isminizi canlandıralım.
- Efektleri değiştirelim.
- Görsel efektleri klavye tuşları ile canlandıralım.

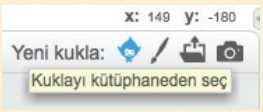
Harflerimizi İçeri Aktaralım

Bu bölümde bir animasyon hazırlayacağız. Bir kelimenin (bu isminiz de olabilir) harflerini canlanlandırıp çeşitli efektler ekleyeceğiz. Biz bu bölüm için, **SCRATCH** kelimesinin harflerini seçtik. Projemize başlamak için, öncelikle yeni bir Scratch projesi açalım (Dosya menüsünün altında Yeni komutunu seçerek yapabiliriz). Ardından Scratch kedimizi silelim. (Şekil-1)



Şekil 1

Şimdi Scratch kütüphanesinde bulunan harflerden S, C, R, A, T, C ve H harflerini programın içine aktaralım. (Şekil-2)



Şekil 2

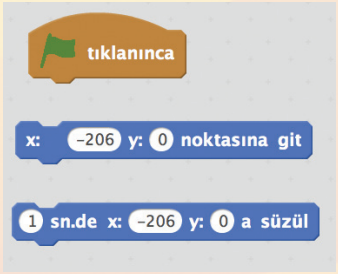
Harfler, şu an dağınık biçimde olacaklardır. (Şekil-3)

Şekil 3

Bu harfleri sürükleyip bırakarak animasyon sonunda olmasını istediğimiz koordinatlara getirelim. (Şekil-4)

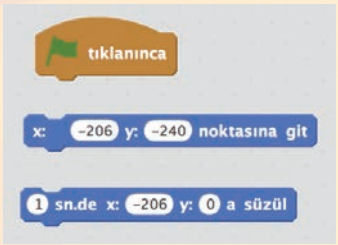
Şekil 4

Şimdi tek tek harflere girip bazı animasyonlar uygulayalım. Scratch üzerinde bir karakteri seçip **Hareket** bölümünden o an bulunduğu koordinatları görebiliriz. Bu sayılar dinamik olarak güncellenmektedir. Öncelikle **S** harfinin aşağıdan yavaşça gelmesini istiyoruz. Bunun için **S** harfini seçip ardından şekildeki komutları kod alanına taşıyalım. (Şekil-5)



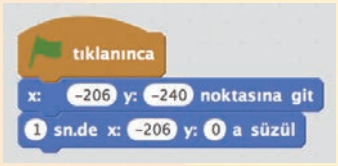
Şekil 5

Şu an gördüğümüz koordinatlar **S** harfinin bulunduğu koordinatlardır. Biz aynı hizada, aşağıdan başlamasını istiyoruz. O halde **y konumunu** en aşağıda yani -240 yapmalıyız. İkinci koordinat yani şu an bulunduğu yer aynen kalmalı. (Şekil-6)



Şekil 6

Son kod bloğunda bulunan **süzül** komutunun süresini değiştirerek en son bulunacağı konuma geleceği süreyi değiştirebiliriz. Kodları birleştirelim. (Şekil-7)

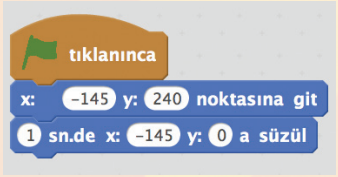


Şekil 7

Şimdi diğer harfimize gelelim. **C** harfi yukarıdan aşağıya doğru gelip sonrada etrafında bir tur dönsün. Yukarıdakine benzer şekilde kod bloklarımızı kodlama alanına getirelim. (Şekil-8)

Şekil 8

C harfinin aynı hizada yukarıda başlaması için **y ekseni** değerini olabilecek en yüksek değer olan 240 yapıyoruz. (Şekil-9)



Şekil 9

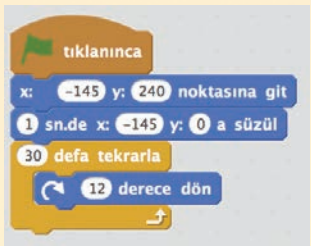
C harfi için, S harfinden farklı olarak kendi etrafına dönme ekleyelim. Bunun için "**... defa tekrar**" ve "**... derece dön**" bloklarına ihtiyacımız olacak.
(Şekil-10)

Şekil 10

Kendi etrafında döndürmek için 360 derecelik bir dönüş ayarlamamız gerekiyor. Bunu kaç defada yapacağımız bize kalmış; eğer 20 defada bu dönüşü yapmak istiyorsak, 360/20 = 18 derecelik dönüşler yapmalıyız. (Şekil-11)

Şekil 11

Eğer bu dönüşü 30 defada tamamlamak isteseydik 360/30=12 derecelik dönüşler yapmamız gerekirdi. **C** harfi için kodumuzu birleştirelim. (Şekil-12)



Şekil 12

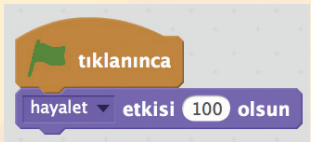
Görsel Efektler Kullanalım

R harfi için biraz daha farklı bir şey yapalım. R harfi yavaştan ortaya belirsin. Bunun için **Görünüm** sekmesinin altında bulunan **hayalet etkisini** kullanacağız. (Şekil-13)

Şekil 13

Hayalet etkisi bir şeklin saydamlığını belirliyor. 100 olması görünmemesini, 0 olması ise resmin orijinal halinin görünmesini sağlıyor.

R harfi için başlangıçta bu etkiyi 100 yaparak R harfinin görünmemesini sağlayacağız. (Şekil-14)



Şekil 14

Ardından 100 defa tekrarla bloğunun içinde **hayalet etkisini -1 artırarak** (yani 1 azaltarak) yavaş yavaş bu etkiyi azaltıp harfin görünmesini sağlayacağız. (Şekil-15)

Şekil 15

Sırada **A** harfimiz var. A harfi belirli bir süre renk değiştirsin istiyoruz. Bunu **görünüm** sekmesinin altında bulunan "**renk etkisini ... arttır**" komutu ile gerçekleştirebiliriz. Scratch'te renk efekti 0 ile 200 arasındadır. Bu nedenle eğer harfimizin aynı renge geri dönmesini istiyorsak toplamda 200 kerelik bir artış yapıp başlangıca dönmeliyiz. (Şekil-16)

Şekil 16

T harfinde farklı bir görsel efekt kullanalım. Kullanacağımız efektin adı **balık gözü**. Önce belirli bir miktar artıralım sonra aynı miktarda azaltalım. Buradaki sayıları değiştirip etkilerini gözlemleyelim. (Şekil-17)

Şekil 17

C harfinin yukarıdan süzülüp gelmesini ve ardından yukarı küçük bir sıçrama yapmasını sağlayalım. (Şekil-18)

Şekil 18

+ BİLGİ

Scratch üzerindeki renkler 0 ile 200 arasındaki sayısal değerlerdir. Bu nedenle eğer bir karakterin renk etkisini 200 artırırsanız karakterin rengi aynı kalacaktır.

Son olarak **H** harfinin kendi etrafında önce saat yönünde ardından saat yönünün tersinde dönmesini sağlayalım. (Şekil-19) Böylece bu bölümdeki ilk uygulamamızı tamamlamış olduk. Siz de istediğiniz kelimelerin harfleriyle farklı animasyonlar yapabilirsiniz. Kullandığımız efektleri ve bunların miktarlarını değiştirerek farklı animasyonları deneyin. Şimdi, birçok uygulamamızın başına koyabileceğimiz bir küçük uygulama yapalım.

Şekil 19

Geri Sayım

Bu bölümde, şimdiye kadar görünüm efektlerini ve hareketi kullanarak animasyon yapmayı öğrendik. Şimdi de ileride uygulamalarımızın çoğunda kullanabileceğimiz herhangi bir uygulama başlamadan önce 3, 2, 1 ve başla gibi bir geri sayım uygulaması geliştirelim.

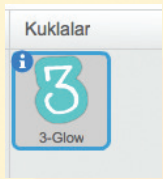
Yeni bir Scratch projesi açıp kedimizi silelim.

Ardından, içeriye yeni bir karakter alalım. Seçeceğimiz karakter **3 sayısı** olsun. "Kuklayı kütüphaneden seç" düğmesine tıklayıp 3 karakterini seçelim. (Şekil-20)

Şekil 20

Bu uygulamada ayrı ayrı karakterler seçmek yerine kılıkları değiştireceğiz.

Şimdi bu karakterimizin ismini değiştirelim. Bunun için, Kuklalar bölümünde bulunan 3 karakterinin sol üst köşesindeki **i** harfine tıklayalım. (Şekil-21)



Şekil 21

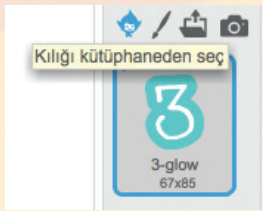
Açılan seçeneklerden karakterimizin ismini **geri sayım** olarak değiştirelim. (Şekil-22)

Şekil 22

Şimdi 2, 1 ve Başla kılıklarını ekleyelim. Bunun için orta bölümde bulunan Kılıklar bölümüne tıklayalım. (Şekil-23)

Şekil 23

Ardından diğer kılıkları eklemek için **Kılığı kütüphaneden seç** komutunu tıklayalım. (Şekil-24)

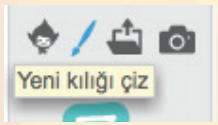


Şekil 24

Ardından 2 rakamını seçelim ve bu işlemi 1 rakamı için tekrarlayalım. (Şekil-25)

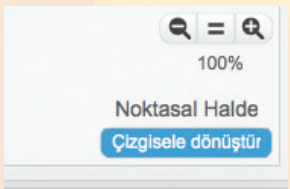
Şekil 25

Son olarak yeni bir kılık çizeceğiz. "Başla" yazan bir yazı yazalım. Bunu **Kütüphaneden kılık seç** düğmesinin yanında bulunan **Yeni kılığı çiz** düğmesine basarak yapacağız. (Şekil 26)



Şekil 26

Karşımıza çoğu bilgisayarda bulunan Paint programına benzer bir arayüz çıkacaktır. Bu arayüzü kullanmanın 2 modu var: Bunlardan biri noktasal, diğeri çizgisel mod. Yazımızı yazmak ve ileride üzerinde daha rahat değişiklik yapabilmek için öncelikle çizgisel moda geçmesini sağlayalım.
(Şekil-27)



Şekil 27

Çizim araçlarının sağ tarafa geçtiğini göreceğiz. Şimdi bu araçlardan Yazı aracını seçelim. (Şekil-28)

Şekil 28

Ekranın ortasına "**BAŞLA**" yazalım (Bazı yazı tipleri Türkçe karakterleri desteklemediği için BASLA olarak yazabilirsiniz). (Şekil-29)

Şekil 29

Başla yazısının üstüne tıklayıp köşelerinden çekerek istediğimiz büyüklüğe getirebiliriz. (Şekil-30)

Şekil 30

Şimdi **Diziler** bölümüne gelip kodlamamızı yapalım. Yapacağımız şey 3, 2, 1 ve Başla kılıklarının peş peşe görünmesi. (Şekil-31)

Şekil 31

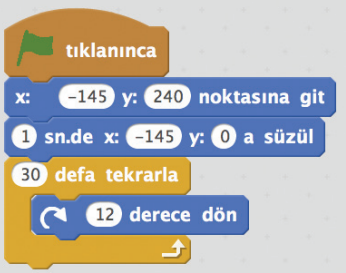
Animasyonumuzu biraz daha süsleyelim. Sayıların ekrandan küçülerek gitmesini sağlayan kod bloklarını inceleyelim.

Şekil 32

BU BÖLÜMDE KULLANILAN KOD BLOKLARI

A-Glow

C-Glow

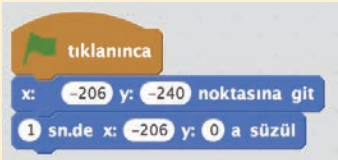


H-Glow

C-Glow2

R-Glow

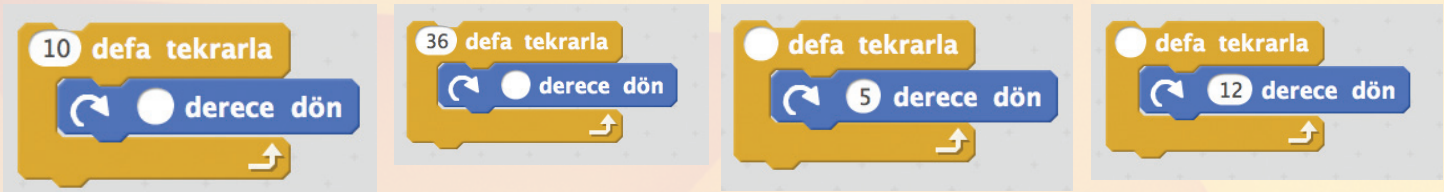
S-Glow



ÇALIŞMA SAYFASI

1. Aşağıdaki kod bloklarını uyguladığımızda hayalet etkisi 0'a indiğine göre kod bloklarındaki boşlukları dolduralım.

2. Bir şeklin kendi etrafında tam bir dönüş yapmasını istiyoruz. Buna göre aşağıdaki kod bloklarında bulunan boşlukları dolduralım.

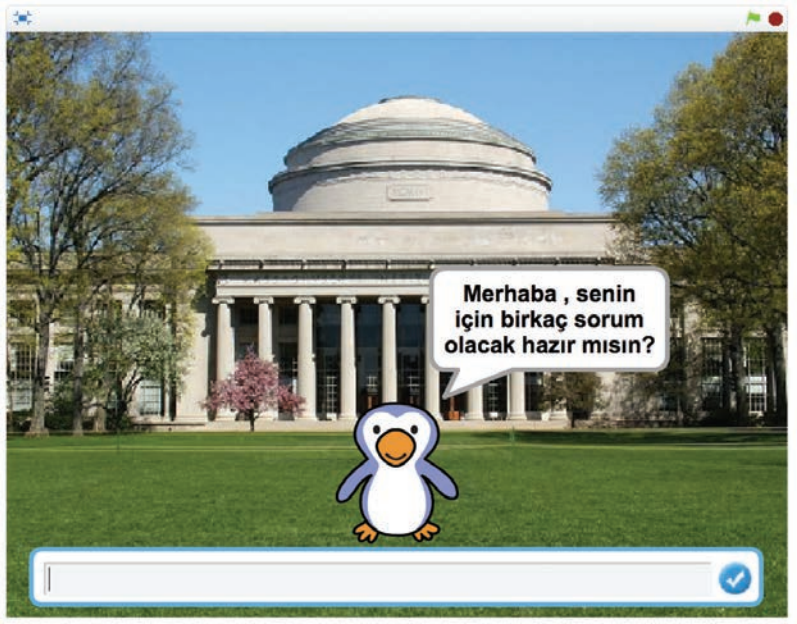


3. Aşağıdaki kod blokları ile bir şekil büyütülüp küçültülmektedir. Kod bloklarının sonunda şeklin başlangıçtaki boyda olması için kod bloklarında bulunan boşlukları dolduralım.

Quiz Uygulaması

Bu Aktivitede Neler Öğreneceğiz?

- Algılama bloklarından sor ve yanıt bekle bloğunu kullanmayı,
- **Veya** bağlacını kullanmayı,
- Kelimeleri birleştirmeyi.



BİLGİ

Scratch programı saniyede 30 kare oynatabilmektedir.

EK GÖREV

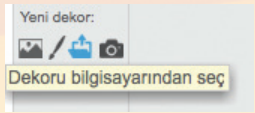
- Soru sayısını artıralım.
- Doğru bilene kadar aynı soruyu sormasını sağlayalım.

Karakterlerimizi Seçelim

Bu bölümde iki tane Quiz uygulaması yapacağız. Bu uygulamalar içinde kullanacağımız arka planları **www.ogreniyorum.xyz** adresinde bulabilirsiniz.

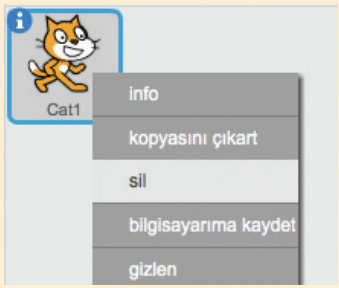
İlk uygulamamızda 3 tane soru soracağız. Verilen cevap doğru ise yarışmacıyı tebrik edeceğiz, yanlış ise cevabın yanlış olduğunu belirteceğiz.

Öncelikle sahnemizi değiştirelim. Bunun için sol altta bulunan **Sahne** bölümünden **Dekoru Bilgisayarımdan Seç** düğmesini tıklayalım. Ardından, arkaplan.png dosyasını içeri aktaralım. (Şekil-1)



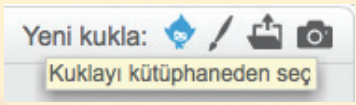
Şekil 1

Sorularımızı, kedimizle sorabileceğimiz gibi başka bir karakter ile de sorabiliriz. Bu uygulama için bir başka karakter seçelim. Bunun için öncelikle, hazırda bulunan kedinin üzerine farenin sağ tuşu ile tıkladığımızda çıkan menüden kedimizi silelim. (Şekil-2)



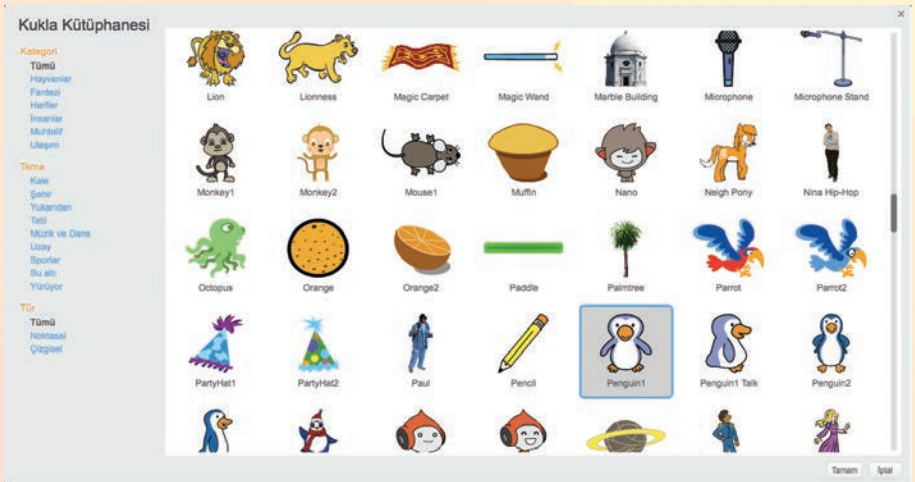
Şekil 2

Şimdi, içeriye soru sormasını istediğimiz karakteri alalım. Bunun için, "Kuklayı kütüphaneden seç" komutuna tıklayalım. (Şekil-3)



Şekil 3

Ardından, kütüphaneden istediğimiz karakteri içeri aktaralım. Biz, Penguin1 **karakterini içeri aktaracağız.**
(Şekil-4)



Şekil 4

Uygulamamızı her başlattığımızda karakterimizin durmasını istediğimiz yeri ve karakterimizin büyüklüğünü kod bloklarıyla ayarlayalım. (Bu ayarları istediğimiz şekilde değiştirebiliriz.) (Şekil-5)

Şekil 5

Bir soru sorup dışarıdan cevap alabileceğimiz kod blokları **Algılama** sekmesinin altında bulunmaktadır. Sorduğumuz bir sorunun cevabının kaydedildiği yer; yine **Algılama** sekmesinin altında, Scratch tarafından oluşturulmuş **Yanıt** değişkenidir. (Şekil-6)

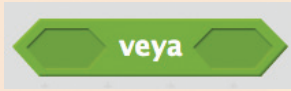
Şekil 6

Sorulamızı Soralım

Kullanıcıya ilk sorumuz, sorularımıza cevap vermek isteyip istemediği olsun. Yanıt eğer **evet** ise yarışmaya başlayacağız, **değilse** "Bir başka sefere" deyip uygulamayı sonlandıracağız. (Şekil-7)

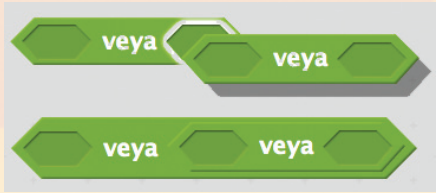
Şekil 7

Scratch programı büyük-küçük harfe duyarlıdır. Yani vereceğiniz yanıtın **Evet, EVET** ya da **evet** olması birbirinden farklıdır. Bu cevaplardan birden çoğunu kabul etmek için **veya** işlemini kullanmalıyız. (Şekil-8)



Şekil 8

Veya işlemi, şekilde de gördüğümüz gibi, 2 tane alana sahip fakat bizim 3 tane cevabımız var. Bunun için iç içe iki tane **veya** işlemi kullanmamız gerekecek. (Şekil-9)



Şekil 9

Verilen tüm bu cevapları doğru kabul etmek istersek kod bloklarımızda şöyle bir değişiklik yapmalıyız:
(Şekil-10)

Şekil 10

Uygulamamızın içinde, 3 tane soru soracağız. Siz, istediğiniz soruları sorabilirsiniz. Bizim soracağımız sorular:

1. Bu uygulama hangi programlama dilinde yazılmıştır?
2. Şu an kullandığımız programlama dili hangi üniversite tarafından yazılmıştır?
3. Scratch programlama dili hangi yılda çıkmıştır?

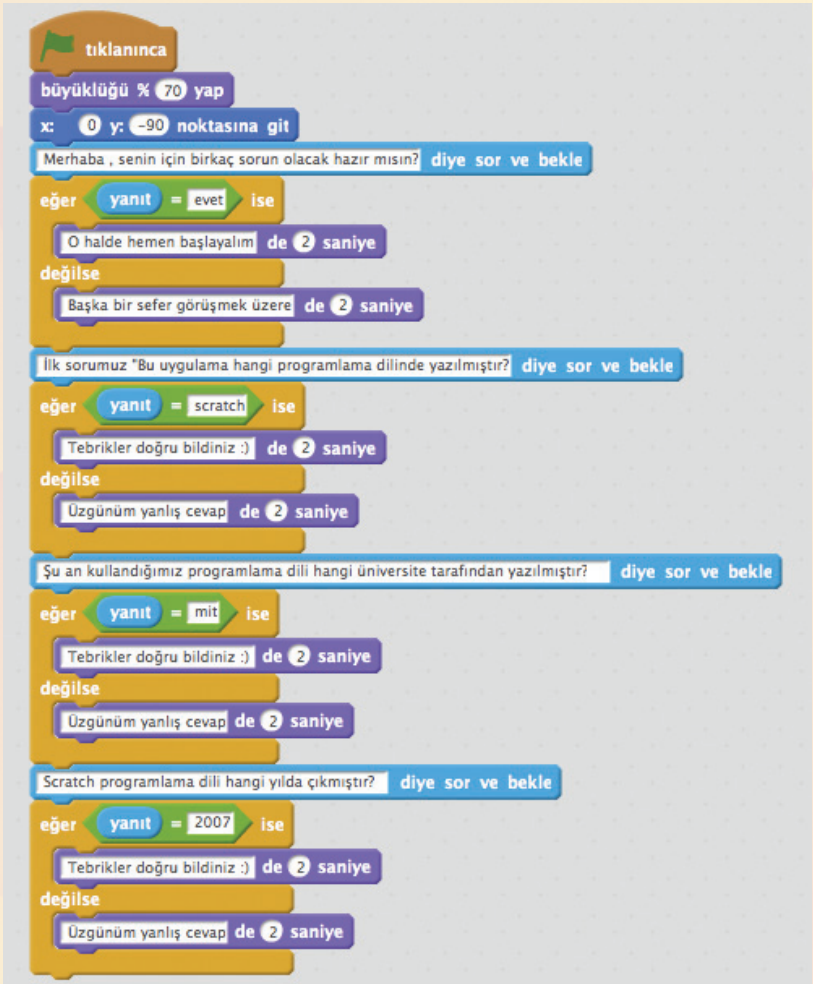
Soracağımız soruların kod blokları şu şekilde olacaktır: (Şekil-11, 12, 13)

Şekil 11

Şekil 12

Şekil 13

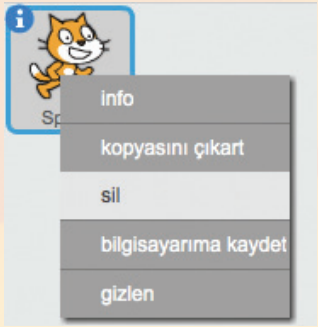
Böylece ilk Quiz uygulamamızı tamamlamış olduk. (Şekil-14)



Şekil 14

Otomatik 4 İşlem Soruları

Bir önceki uygulamamızı biraz geliştirelim. Bu uygulamamızda, bize otomatik olarak toplama işlemi soran bir uygulama yapacağız. Yeni bir proje açıp uygulamamıza başlayalım. Öncelikle kedimizi silelim. (Şekil-15)

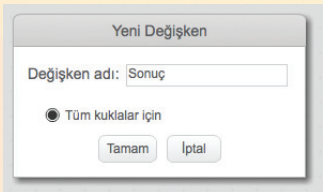


Şekil 15

Bunun için, başlangıçta 3 tane değişkene ihtiyacımız var. Bu değişkenler **Sayı1**, **Sayı2** ve **Sonuç** değişkenleri olacak. Bu değişkenleri oluşturalım. (Şekil-16, 17, 18)

Şekil 16

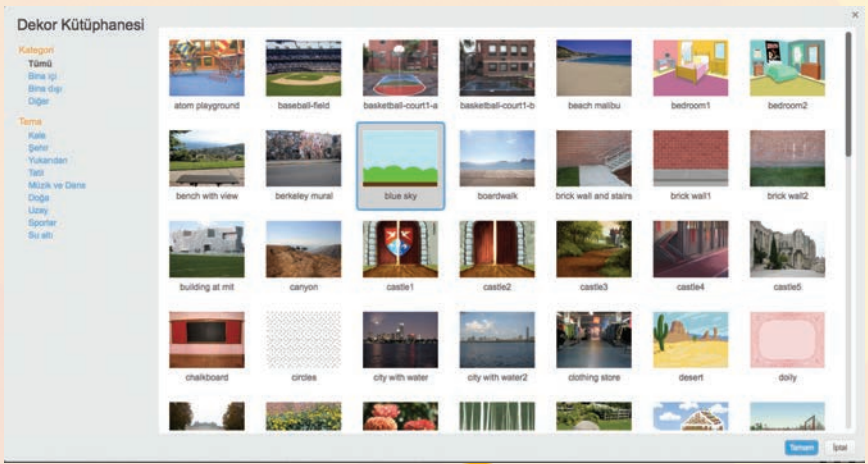
Şekil 17



Şekil 18

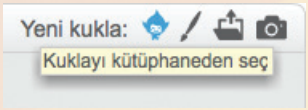
Sorularımızı soracak karakteri seçmeden önce bir arka plan belirleyelim. Bunun için, sol altta bulunan Dekor bölümüne geçip kütüphaneden (ya da dilediğiniz herhangi bir yerden) arka plan seçelim. Dilediğiniz arka planı seçebilirsiniz. (Şekil-19, 20)

Şekil 19

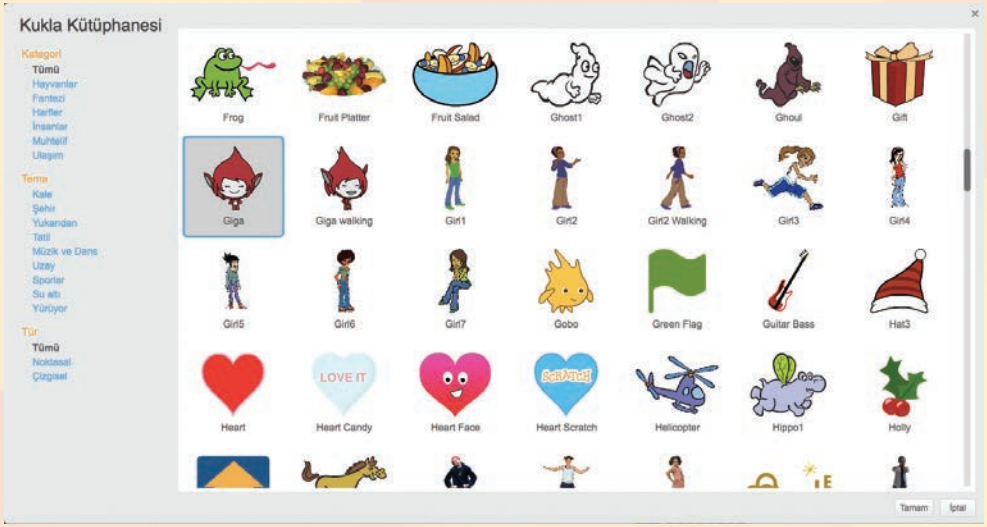


Şekil 20

Şimdi soruları soracak birine ihtiyacımız var. Bu karakteri Scratch kütüphanesinden seçelim. Biz, bu uygulama için **Giga** karakterini seçtik. (Şekil-21, 22)



Şekil 21



Şekil 22

Karakterimizi, durmasını istediğimiz yere alıp uygulamamızı yazmaya başlayalım. (Şekil-23)

Şekil 23

Sayı1 ve Sayı2 adlı değişkenlerin rastgele sayılar olmasını istiyoruz. Scratch programında rastgele sayılar üretmek için **İşlem** sekmesinin altında bulunan "**1 ile 10 arasında bir sayı (tut)**" komutunu kullanmamız gerekiyor. (Şekil-24)

Şekil 24

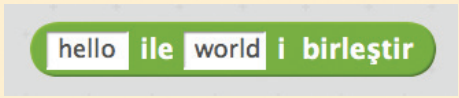
Sayı1 ve Sayı2 değişkenleri 1 ile 10 arasında rastgele sayılardan oluşacak (Dilerseniz bu aralığı değiştirebilirsiniz).
(Şekil-25)

Şekil 25

Sonuç ise Sayı1 ve Sayı2'nin toplamı olacak. Toplama işlemini gerçekleştirmek için **İşlem** sekmesinin altında bulunan toplama işlemini kullanacağız. (Şekil-26)

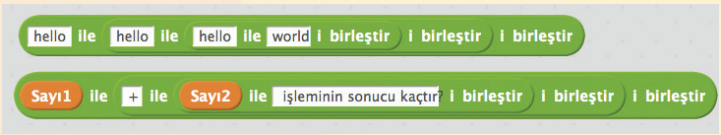
Şekil 26

Peki oyunumuzun bize bu sayıları otomatik olarak sormasını nasıl sağlayacağız? Bunu da yine **İşlem** sekmesinin altında bulunan bir komutla gerçekleştireceğiz. (Şekil-27)



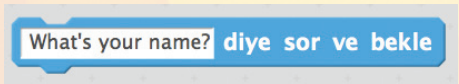
Şekil 27

Kelimeleri birleştirmemizi sağlayan bu komut sadece iki değer alıyor. Bizim sormak istediğimiz cümle ise "**Sayı1 + Sayı2 işleminin sonucu kaçtır?**". Burada "Sayı1+Sayı2" ve "işleminin sonucu kaçtır?" diye 4 ayrı kelimemiz var. Bunun için, iç içe komut blokları kullanmalıyız. (Şekil-28)



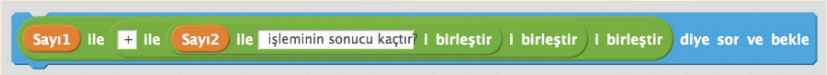
Şekil 28

Soruyu sormak için **Algılama** sekmesinin altında bulunan "**... diye sor ve bekle**" kod bloğunu kullanacağız. (Şekil-29)



Şekil 29

Üstte bulunan **birleştir** komutu ile Algılama sekmesinin altında bulunan "**... diye sor ve bekle**" kod bloğunu birleştirelim. (Şekil-30)



Şekil 30

Sorduğumuz soruya doğru cevap verilirse yani **yanıt, sonuç** ile aynı değerse "Tebrikler Doğru Bildin :)" mesajını, değilse "Üzgünüm Yanlış Cevap" mesajını vereceğiz.
(Şekil-31)

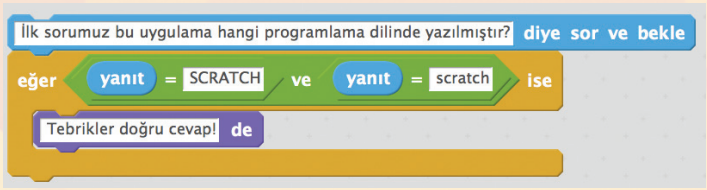
Şekil 31

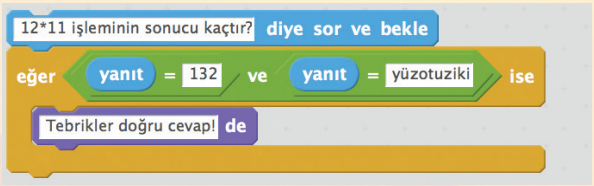
Tüm bunları bir araya getirelim. Ve kaç tane soru sormak istiyorsak, kodu böyle bir döngü içine alalım. (Şekil-32)

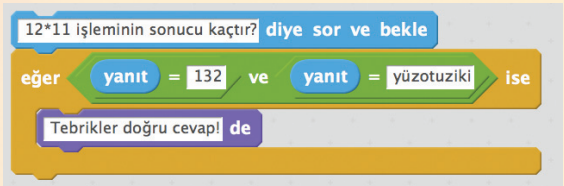
Şekil 32

ÇALIŞMA SAYFASI

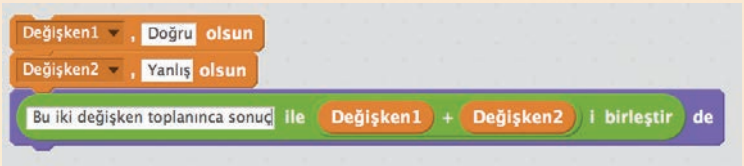
1. Scratch ile program yazmayı çok seven Kuzey, verilen bir soruya girilebilecek birden çok cevabı kabul eden bir kod bloğu yazmaktadır. Fakat aşağıdaki kod blokları istediği gibi çalışmamaktadır. Kuzey'e kod bloklarını düzeltmekte yardımcı olur musunuz?







2. Aşağıda kod bloklarında bazı kelimeler birleştirilmiştir. Bu cümlelerin tamamını yazar mısınız?

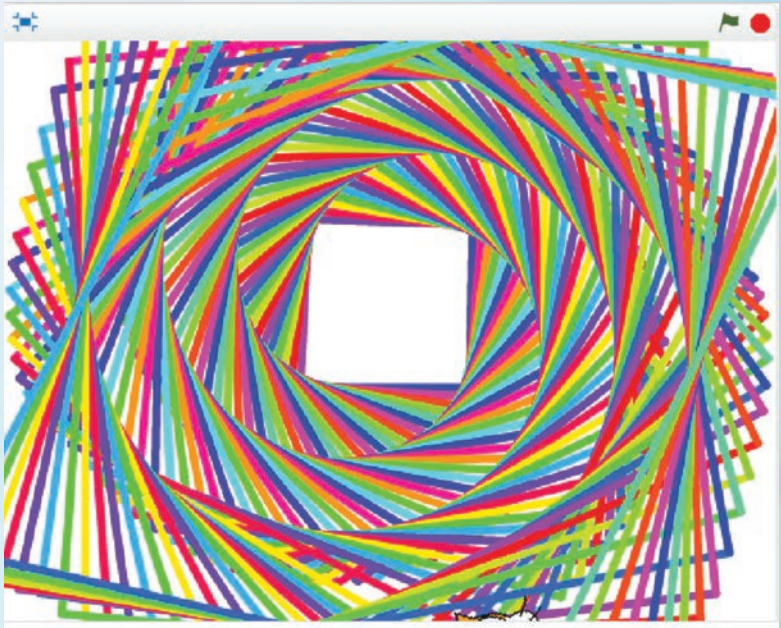


3. Aşağıda verilen işlemlerden elde edilebilecek en büyük ve en küçük tam sayı değerleri bulunuz.

Kaplumbağa Geometrisi

Bu Aktivitede Neler Öğreneceğiz?

- Kaplumbağa geometrisini,
- Kalem sekmesini kullanmayı,
- Kalem efektleri kullanmayı,
- Düzgün çokgen çizmeyi.



BİLGİ

Scratch günü her yıl çeşitli aktiviteler düzenlenerek kutlanmaktadır. Size en yakın aktiviteyi **http://day.scratch.mit.edu/** adresinden bulabilirsiniz.

EK GÖREV

- Farklı çokgenlerle farklı şekiller deneyelim.
- Çember çizmeye çalışalım.
- Kalemin kalınlığını program içerisinden değiştirelim.

Düzgün Çokgenler Çizelim

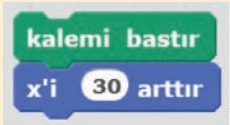
Öncelikle, kedimizi silip Scratch kütüphanesinde bulunan kalem karakterlerinden birini seçelim. (Şekil-1)

Şekil 1

Seçtiğimiz kalem karakterini programımıza yükleyelim. (Şekil-2)

Şekil 2

Ardından, kalem sekmesini kullanarak küçük bir doğru parçası oluşturalım. (Şekil-3)

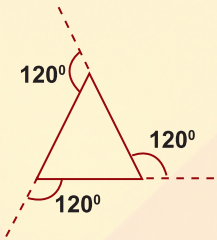


Şekil 3

Kalemi bastırdığımız müddetçe, her yaptığımız ilerlemede kalem çizgi çizmeye devam edecektir. Kalemi kaldırdığımız andan itibaren yaptığımız hareketlerde, ekranda herhangi bir değişiklik olmayacaktır. Kesik kesik çizgi çizecek bir kod bloğu oluşturalım. (Şekil-4)

Şekil 4

Elimizdeki kalemle ekranda bir eşkenar üçgen oluşturmaya çalışalım. Bir eşkenar üçgen çizmek için, önce bir miktar çizgi çizip 120 derece dönmeliyiz. Daha sonra aynı uzunlukta bir çizgi daha çizip 120 derece daha dönmeliyiz. Bu işlemi toplamda üç kez tekrarladığımızda başlangıca dönmüş oluruz. Dönme açılarımız, aslında dış açıları belirliyor ve dış açılar toplamı 360 derece olduğu anda başlangıç noktasında dönüyoruz. (Şekil-5)



Şekil 5

Ayrı ayrı üçer defa yazmak yerine, "**100 adım git**" ve "**120 derece dön**" kod bloklarını "**üç defa tekrarla**" komutunun içine alacağız. (Şekil-6)

Şekil 6

Bu şekilde bir kare çizelim. Daha sonra beşgen, altıgen ve sekizgen çizelim. Her defasında dış açıları yani dönme açılarını bulalım. 7, 8, 9 ve 10 numaralı şekillerde sırasıyla kare, beşgen, altıgen ve sekizgen için gerekli kod bloklarını görebilirsiniz.

Şekil 7

Şekil 8

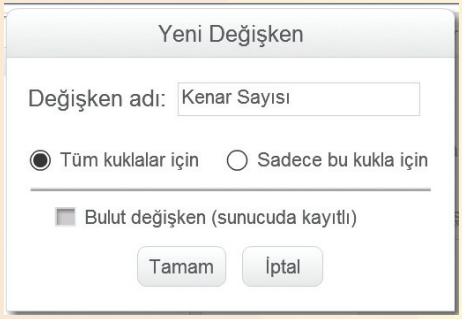
Şekil 9

Şekil 10

Sahnede biriken çizimlerimizi temizlemek için kalem sekmesinin altındaki temizle komutunu kullanacağız. Boşluk tuşuna her bastığımızda ekranın temizlenmesini sağlayalım. (Şekil-11)

Şekil 11

Her çokgen için ayrı ayrı kod yazmak oldukça uzun ve gereksiz bir iş. Bunun yerine, dışarıdan kaç kenarlı bir çokgen çizmek istediğimizi yazdığımız ve buna göre çokgen çizen bir uygulama geliştirelim. Öncelikle, kenar sayısının değerini saklamak için "kenar sayısı" adlı bir değişken oluşturalım. (Şekil-12)



Şekil 12

Bu değişkeni programın içerisindeyken kontrol edip değiştirmek istiyoruz. Bunun için, değişkenimizi oluşturduktan sonra değişkenin üzerine gelip faremizin sağ tuşuna bastığımızda çıkan menüden "sürgü" seçeneğini seçelim. (Şekil-13)

Şekil 13

Sürgü seçeneği sayesinde, artık programın içerisindeyken "kenar sayısı" adlı değişkenimizi değiştirebileceğiz. Değişkenimizin hemen altındaki sürgüyü sağa sola çekerek değişkenimizi artırıp azaltabiliyoruz. (Şekil-14)

Şekil 14

Değişkenimize, yine farenin sağ tuşu ile bastığımızda en büyük ve en küçük değişkenleri belirleyebiliyoruz. Bu kısım oldukça önemli, özellikle uç durumların oluşması durumunda. Örneğin; sıfırgen, birgen ya da ikigen diye şekillerimiz yok. Bu sebeple, değişkenin alt limitini üçten başlatmazsak program içerisinde bir sorun çıkacaktır. O nedenle kodlamaya başlamadan önce değişkenimizin alt ve üst limitlerini belirleyelim. Alınabilecek en küçük değer 3 olsa da en yüksek değer ise size kalmış. Biz 15 olarak belirleyeceğiz. (Şekil-15)

Şekil 15

Şimdi, bir çokgeni nasıl çizeceğimizin algoritmasını basitçe anlamaya çalışalım.

1. Kenar sayısı belirlenir (Bunu sürgü ile belirleyeceğiz).
2. Kenar sayısı kadar tekrar edilecek bir döngü oluşturulur.
3. İstenilen kenar uzunluğu kadar bir çizgi çizilir.
4. Dönme açısı (360/kenar sayısı) kadar dönülür.

Klavyede herhangi bir tuşa basılınca yukarıda söylenenleri gerçekleştirecek bir kod bloğu yazalım. Bu kod bloğunun tetikleyicisi olarak herhangi bir harfi belirleyebilirsiniz. Biz "a" harfini seçtik. Kenar sayısı değişkenini, sürgüyü kullanarak çeşitli sayılara getirip "a" tuşuna basarak birkaç tane çokgen çizelim. (Şekil-16)

Şekil 16

Daha sonra kalemimizi belirli bir açı kadar çevirecek şekilde bir tuş atayalım. Biz 15 derece çevirecek şekilde "s" tuşuna atıyoruz. (Şekil-17)

Şekil 17

Belirli bir kenar sayısını seçip ardından s-a-s-a-s-a-s-a-s-a… tuşlarına basarak çeşitli şekiller oluşturalım.

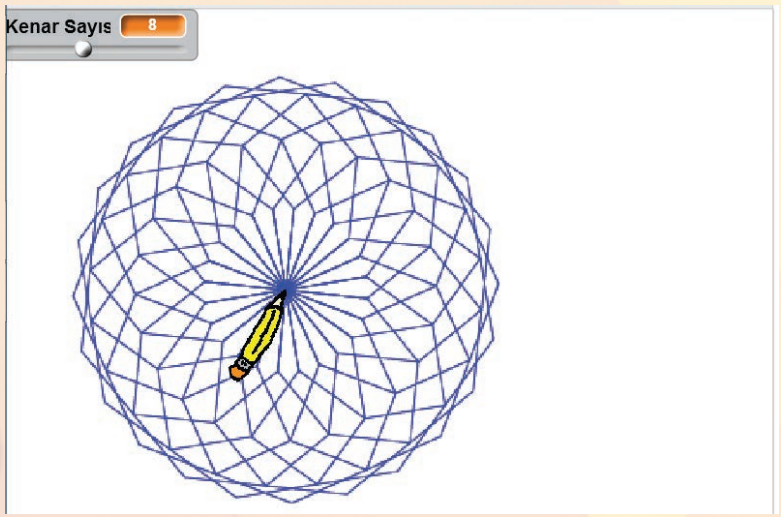
Bu şekilde başlangıç noktasına dönebilmek için, 360/15=24 kere "s" tuşuna basmamız gerekiyor.

Kaplumbağa ile Sanat

"d" tuşuna bastığımızda 24 kere çokgen çizip her çokgen çizdikten sonra 15 derece dönecek şekilde bir kod bloğu oluşturalım. (Şekil-18)

Şekil 18

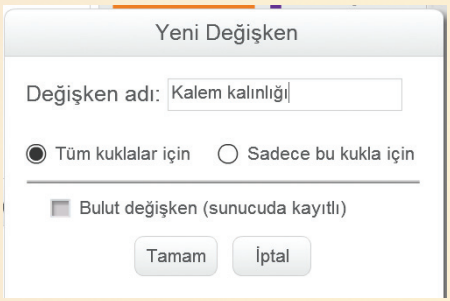
Kenar sayısı 8 iken yapılmış olan bir çizimi aşağıda bulabilirsiniz. (Şekil-19)



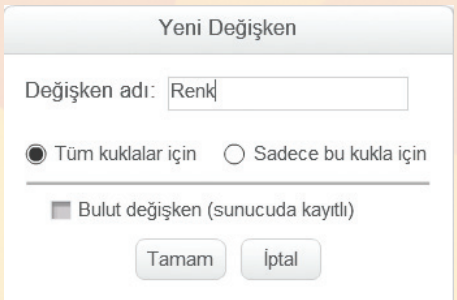
Şekil 19

Bir modeli tamamlamak için şeklin 360 derece dönmesi gerekmektedir.

Kalemin kalınlığı ve rengi gibi şeyleri değiştirerek çok daha güzel şekiller oluşturabiliriz. Öncelikle, kalemin kalınlığını ve rengini bir değişkene bağlı olarak (kenar sayısı gibi) değiştirmeyi sağlayalım. Kalem kalınlığı ve renk adlı iki değişken oluşturalım. (Şekil-20, 21)



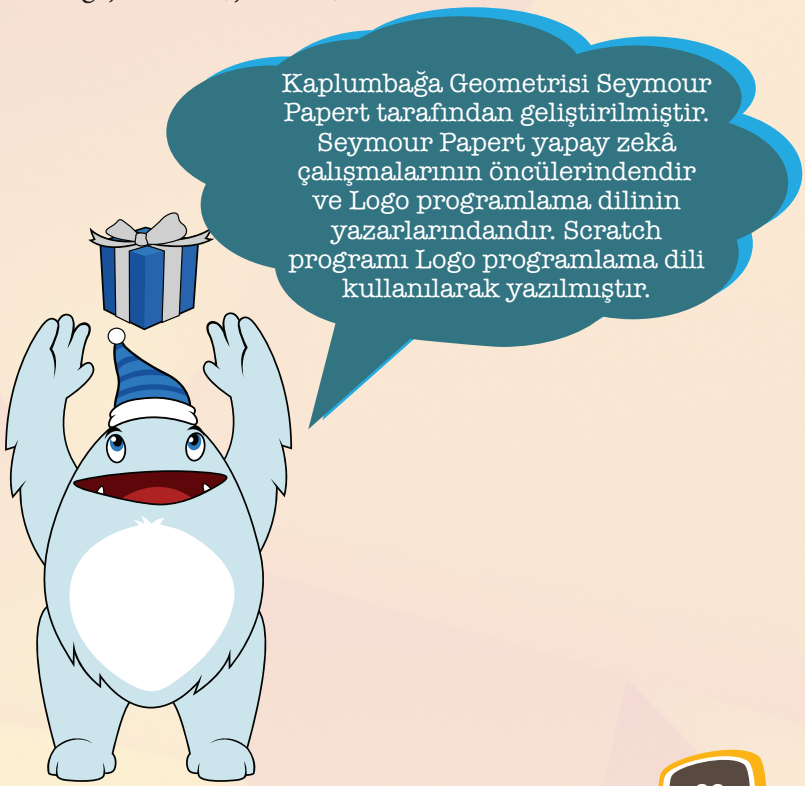
Şekil 20



Şekil 21

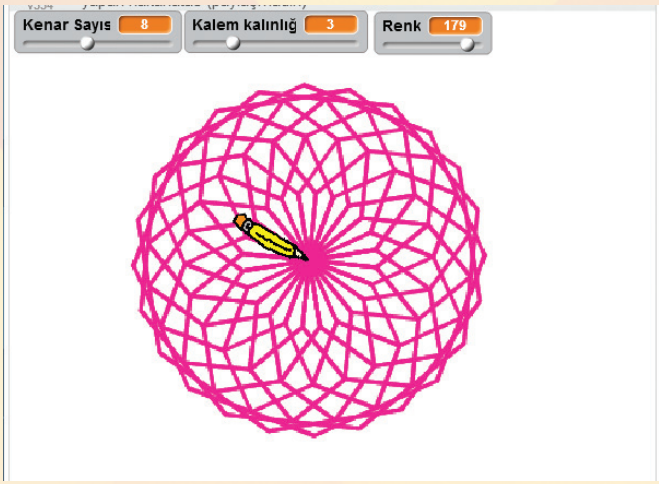
Kalem kalınlığını 1'den 10'a, renk değişkenini ise 0'dan 200'e kadar üst değerlerle oluşturalım.

Kalemimizin rengini ve kalınlığını bu değişkenlere bağlı olarak değiştirelim. (Şekil-22)



Şekil 22

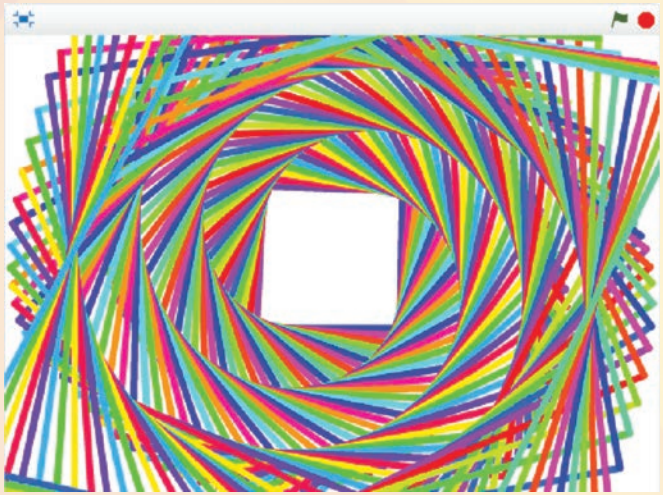
Renkler ve kalem kalınlığıyla oynayarak birkaç süsleme yapalım. (Şekil-23)



Şekil 23

Bundan sonra kodları değiştirerek farklı şekiller çıkarmak size kalmış. Biz, bir iki farklı örnek daha verip bu bölümü bitirelim.

Şekil 24'deki gibi bir çizimi nasıl elde edebileceğimizi görelim.



Şekil 24

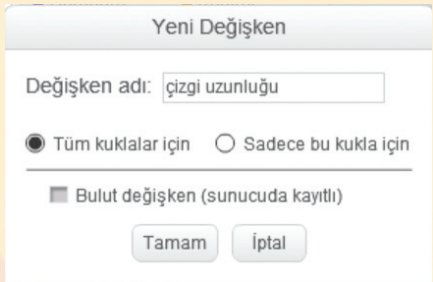
BİLGİ

Temizle kod bloğuyla kalem sekmesi altında yaptığınız tüm çizimleri bir anda silebilirsiniz.

Birkaç küçük değişiklik yapalım ve neler elde edebileceğimizi görelim. Öncelikle, yeşil bayrak tıklandığında başlangıç değerlerimizi oluşturalım. Kalemi kaldırıp merkez olarak istediğiniz bir noktaya gitmesini (biz (-50,-35) noktasını seçtik) sağlayalım. Ardından kalemin kalınlığını ayarlayıp ekranı tamamen temizleyelim. (Şekil-25)

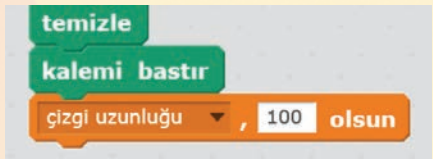
Şekil 25

Dikkat ettiyseniz, yukarıdaki çizimde çizgilerimizin uzunluğu farklı. Çizgi uzunluğunu sürekli olarak değiştireceğimiz için, "çizgi uzunluğu" adlı bir değişken oluşturalım. (Şekil-26)



Şekil 26

Başlangıç çizgimizin uzunluğunun kaç olmasını istiyorsak değişkenimize o değeri atayalım. Biz, bu değeri 100 olarak belirledik. (Şekil-27)



Şekil 27

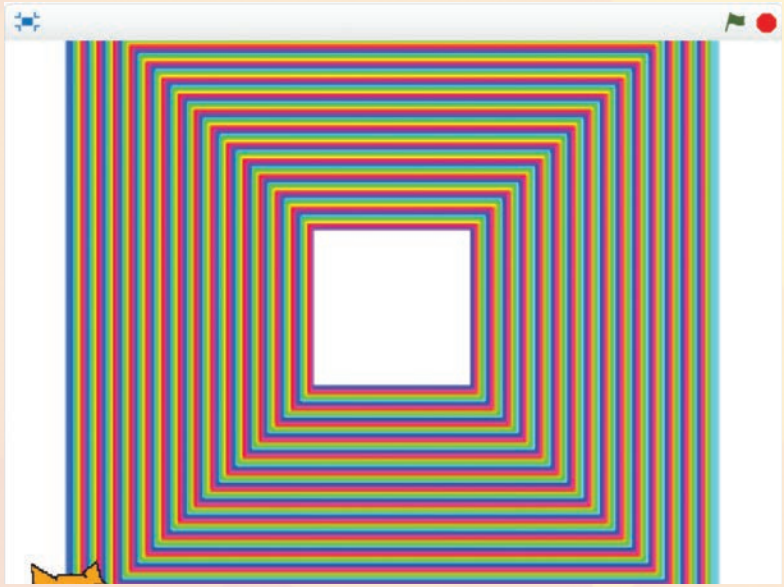
BİLGİ

Bir çokgenin kenar sayısını artırdıkça şeklinin çembere yaklaştığını fark edebilirsiniz.

Ardından, kaç kere çizgi çizileceğini belirten bir döngü oluşturalım. Her defasında kalemin rengini değiştirirken çizgimizin uzunluğunu artıralım. Her çizgiden sonra belirli bir derece dönmesini sağlayalım. Öncelikle, kod bloğumuzu dönüşü 90 derece olacak şekilde düzenleyelim. (Şekil-28)

Şekil 28

Bu kod sonucu oluşacak grafik nasıl bir şey olabilir? (Şekil-29)



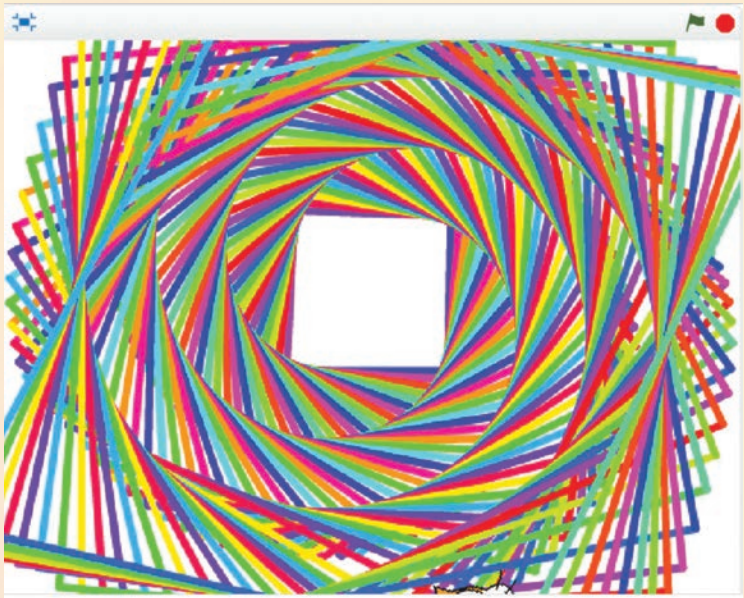
Şekil 29

Bir önceki kod satırında (Şekil-30) yapacağımız çok küçük bir değişiklikle neler oluşturabileceğimize bir bakalım.

90 derecelik dönüş açısını 89 derece olarak düzeltelim. Yeşil bayrağa tıklayalım.

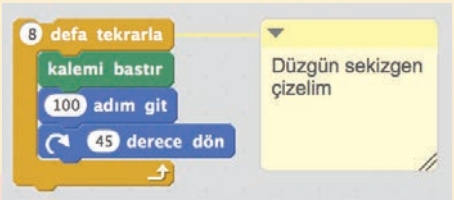
Şekil 30

1 derecelik fark ile elde ettiğimiz farka dikkat edin, siz de farklı dereceler ve uzunluklar ayarlayarak kendi denemelerinizi yapabilirsiniz. (Şekil-31)



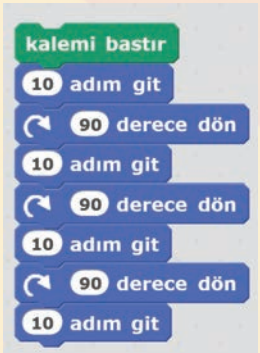
Şekil 31

BU BÖLÜMDE KULLANILAN KOD BLOKLARI



ÇALIŞMA SAYFASI

1. Aşağıda verilen kodlar ile neler yapılabileceğini tahmin etmeye çalışalım.



2. Aşağıdaki kod bloklarına göre bu tuş dizinleri basıldığında oluşabilecek şekilleri çiziniz.

   a. a-s-s-s-s-a-s-s-s-s-a-s-s-s-s

   b. a-a-s-s-a-a-s-s-a-a-s-s-a-a-s-s-a-a-s-s-a-a-s-s

   c. Bir kare çizmek için a ve s tuşu kombinasyonu ne olmalıdır?

   d. d-d

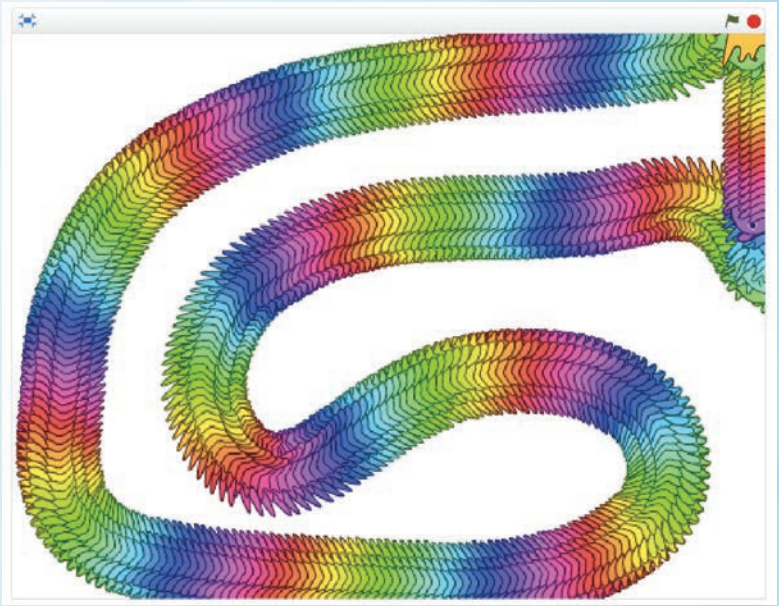
   e. d-d-d

   f. d-d-s-s-d-d

Kalem Sekmesi

Bu Aktivitede Neler Öğreneceğiz?

- Kalem komutlarını tanıyalım,
- Koordinat düzlemi,
- Harfi kalem ile çizmek,
- Fare ile çizim yapmak,
- İz bırak aracını kullanmak.



BİLGİ

Scratch 2.0 içerisine v435.2 ile transparan kalem özelliği eklenmiştir.
Daha detaylı bilgi için; **http://wiki.scratch.mit.edu/wiki/Pen** adresine göz atabilirsiniz.

EK GÖREV

- Kalem sekmesini kullanarak kendi adınızı yazmayı deneyelim.
- Kalem sekmesini kullanarak yüz çizmeyi deneyelim.
- İz bırak kod bloğunu kullanarak sürekli uzayan bir yılan karakteri yapalım.

- **temizle** → Kalem ve iz bırak aracıyla yapılan tüm çizimleri temizler.
- **iz bırak** → Seçili karakterin sahne üzerine izini bırakır.
- **kalemi bastır** / **kalemi kaldır** → Kalemin bastırılmasını, böylece karakter hareket ettikçe çizim yapılmasını sağlar.
- **kalem rengini yap** → Kalemin rengini seçtiğimiz şekilde ayarlar.
- **kalem rengini 10 arttır** → Kalemin rengini belirli miktarda artırır.
- **kalem rengini 0 yap** → Kalemin rengini belirtilen sayı olarak ayarlar.
- **kalem tonunu 10 arttır** → Kalemin renk tonunu belirlenen miktarda artırır.
- **kalem tonunu 50 yap** → Kalemin renk tonunu belirtilen sayı olarak ayarlar.
- **kalem kalınlığını 1 arttır** → Kalemin kalınlığını belirtilen miktarda artırır.
- **kalem kalınlığını 1 yap** → Kalemin kalınlığını belirtilen sayı olarak ayarlar.

Şekil 1

Sadece kalem sekmesi kullanarak oyunlar yapabileceğinizi biliyor muydunuz? https://scratch.mit.edu/projects/75258684/ adresini ziyaret etmelisiniz.

Fare İmleciyle Çizim Yapmak

Scratch Kalem aracını kullanmak için, elimizde bir kalem olduğunu ve Scratch üzerinde bulunan sahnenin bir kağıt olduğunu hayal edelim. Eğer kalemi bastırmazsak, elimizi istediğimiz kadar hareket ettirelim yazı yazamayacağız. Fare imlecimizi takip ederek çizim yapan bir program yazalım. Tabii çizimimizi elimizi bastırdığımız sürece yapabileceğiz.

Ekranda çizim yapmamızı engelleyen herhangi bir nesne olmasını istemiyoruz. Bu yüzden kedimizi gizleyeceğiz. (Şekil-2)

Şekil 2

Ardından, fare imlecimiz nerede ise karakterimizin oraya gitmesini sağlayalım. Bu, karakterimizin sürekli fare okunun bulunduğu yere gitmesiyle mümkün olabilir. (Şekil-3a, b, c, d)

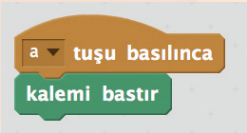
Şekil 3a

Şekil 3b

Şekil 3c

Şekil 3d

Yeşil bayrağa tıkladığımızda kedimiz kaybolacak ve fare imlecimiz ekranda boş boş dolaşacak. Kalemi kaldırıp indirmek için iki tane tuşa ihtiyacımız var. Biz bunları A ve S tuşu olarak seçtik. Kalem; A tuşuna bastığımızda kalkmış, S tuşuna bastığımızda inmiş olacak. (Şekil-4a, b)



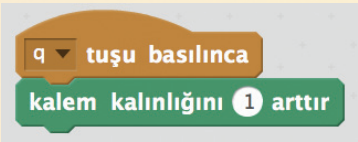
Şekil 4a

Şekil 4b

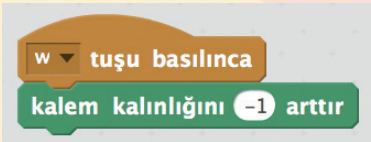
Tek tuşla bunu nasıl yapardık? Örneğin; A tuşuna bastığımız anda kalem basılıysa kaldıralım ya da kalem basılı değilse bastıralım gibi bir kod bloğunu nasıl yazabilirdik?

Şu anda, yeşil bayrağa bastığımızda fare imlecini takip eden görünmeyen bir karakter olacaktır. Klavyemizden **A** tuşuna basarsak kalemi bastıracak yani yazı yazabileceğiz. **S** tuşuna bastığımızda kalemimizi kaldıracak yani yazı yazamayacağız.

Şimdi dilerseniz, kalemin kalınlığı ve rengi ile ilgili düzenlemeler yapalım. Öncelikle, kalemin kalınlığını artırıp azaltmaya yarayan komut bloklarını deneyelim. Bunun için, klavyemizdeki **Q ve W** tuşlarını kullanalım. Kullandığımız kalemin kalınlığını Q tuşu ile artırırken W ile azaltalım. (Şekil-5a, b)

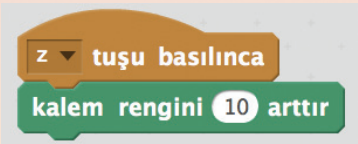


Şekil 5a

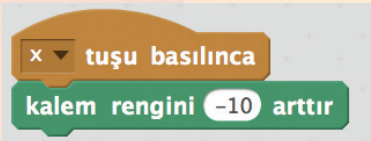


Şekil 5b

Birkaç özellik daha eklemek istersek **Z ve X** tuşlarına bastıkça kalemin rengini değiştirebiliriz. (Şekil-6a, b)



Şekil 6a



Şekil 6b

Biraz karalama yaptıktan sonra ekranı temizleme ihtiyacı duyabiliriz. Ekranı tamamen temizlemek için, kalem sekmesinin altında bulunan **Temizle** kod bloğunu kullanmamız yeterlidir. Dilerseniz, boşluk tuşuna basılınca ekranı temizleyen bir kod bloğu yazalım. (Şekil-7)

Şekil 7

Peki ya sadece ekranın bir bölümünü silmek isteseydik? Yani bir silgimiz olsun isteseydik bunu yapabilir miydik? Cevabı elbette evet ama tam anlamıyla bir silgi yapmak bu bölümün biraz dışında bir konu. Biz basit bir silgi yapacağız ve silgiyi kullandıktan sonra kalemimizin rengini otomatik olarak yeşil yapacağız. Aslında tam olarak bir silgi yapmayacağız, ekranın renginde boyama yapacağız. (Şekil-8)

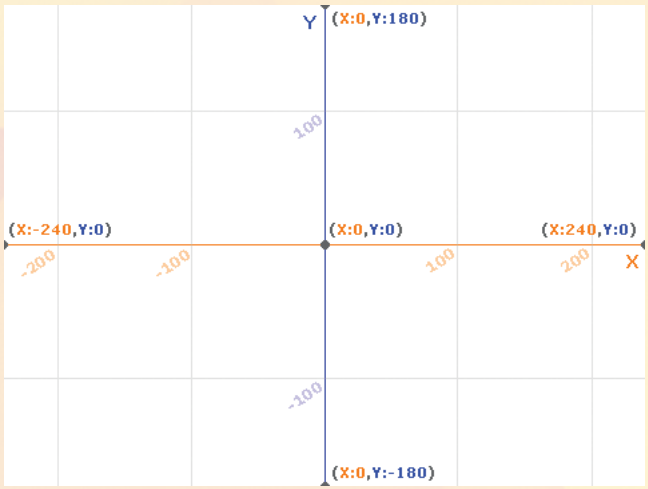
Şekil 8

E tuşu basılınca kalemin basılı olduğundan emin olup kalemin rengini beyaz yapıyoruz ve böylece **E** tuşu basılı olduğu müddetçe fare imlecinin dolaştığı yerleri zeminin rengine boyuyoruz. **E** tuşundan elimizi çektiğimizde ise kalemin rengini yeşil (ya da istediğimiz başka bir renk) yapıyoruz ki çizmeye devam edebilelim.

Silgiyi kullandıktan sonra kalemin bir önceki rengine dönmesini nasıl sağlayabiliriz?

Koordinat Düzlemi

Eğer rastgele şekiller çizmeyeceksek, örneğin tam olarak bir harfi çizmek istiyorsak, biraz matematik bize yardımcı olacaktır. Bu bölümde ekrana kocaman bir A harfi çizeceğiz fakat öncesinde biraz matematiksel kavramları hatırlayalım (ya da öğrenelim). Matematik dünyasında 2 boyutlu dünyayı algılayabileceğiniz birçok basit düzenek var. Birbirine dik olarak çakıştırılmış iki doğru ile tüm pozisyon belirleme sorunlarımızı çözebiliriz. (Şekil-9)

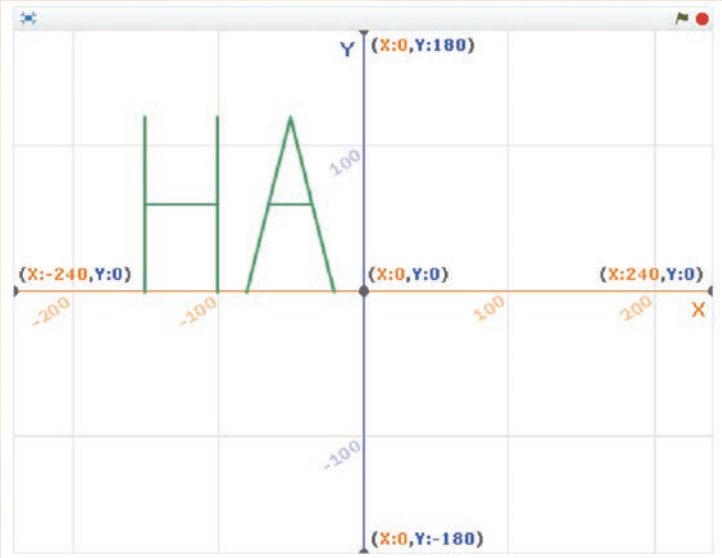


Şekil 9

Bu iki sayı doğrusuna eksen diyoruz. Sağa sola gidene (yatay olan) X ekseni, yukarı aşağı gidene (dikey olan) Y ekseni adını veriyoruz. Siz dilerseniz başka isimler de verebilirsiniz ama bu vakitten sonra kabul görür mü bilemeyeceğiz.

Bir noktanın koordinatlarını (pozisyonunu) anlatmak için X ve Y'leri sırasıyla söyler ve virgülle ayırarak yazarız (X, Y). Örneğin (50, 80) diye bir koordinat görmüşsek bu iki doğrunun çakıştıkları noktadan (orijin) 50 birim sağda ve 80 birim yukarıda olduklarını gösterir. Ya da (-100, 50) noktası orijinden 100 birim solda ve 50 birim yukarıda olduğumuz anlamına gelir.

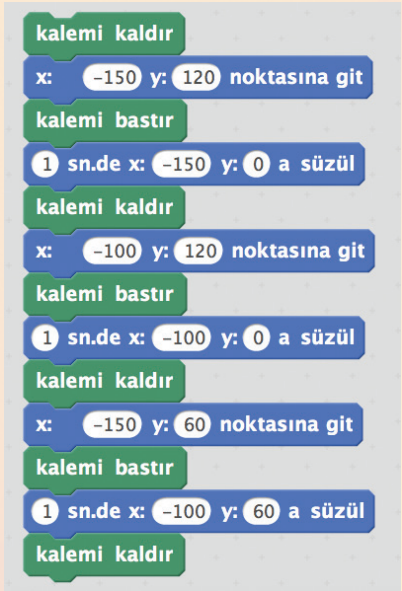
Şekil üzerinde birkaç örnek inceleyelim;



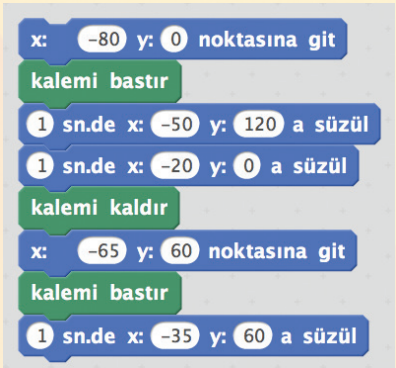
Kalem ile Harf Çizmek

Şekildeki **H ve A** harfini çizmek için gerekli kod bloğunu şu şekilde oluşturabiliriz. (Şekil-12a, b)

Siz de farklı harfleri çizebilirsiniz. Adınızı çizerek yazmayı deneyebilirsiniz.



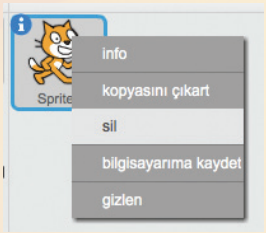
Şekil 12a



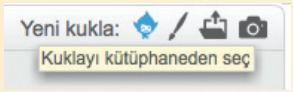
Şekil 12b

İz Bırak Aracını Kullanmak

Kalem sekmesinin en önemli araçlarından biri de **İz Bırak** kod bloğudur. İz Bırak kod bloğunu kullanırken, karakterimizin kılığı neyse onun izini bırakarak hareket edeceğimiz için bu kılık çok önemlidir. İz bırak ile küçük bir uygulama yapacağız. Öncelikle, Gobo karakterini içeri aktaralım. (Şekil-13, 14, 15)



Şekil 13

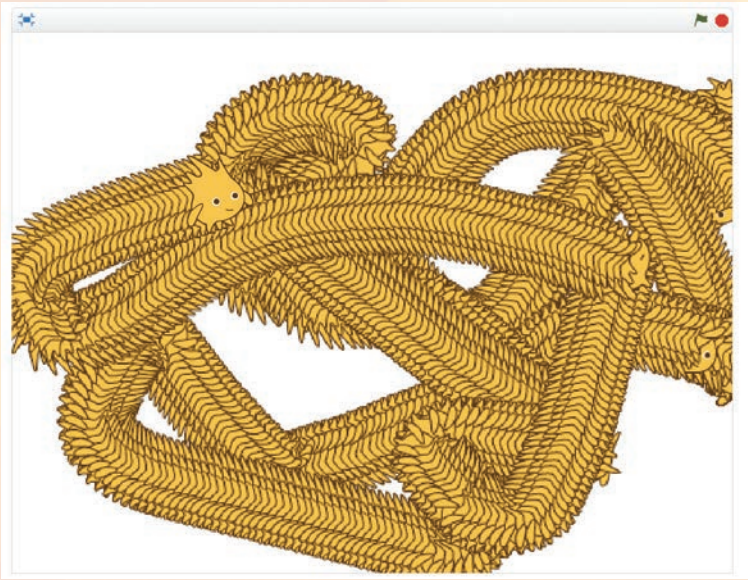


Şekil 14

Şekil 15

Kitabın 6. bölümündeki Kovalamaca uygulamasında yaptığımız fare takibini yapıp Gobo'unun fareyi takip etmesini ve bu sırada iz bırakmasını sağlayabiliriz. (Şekil-16, 17)

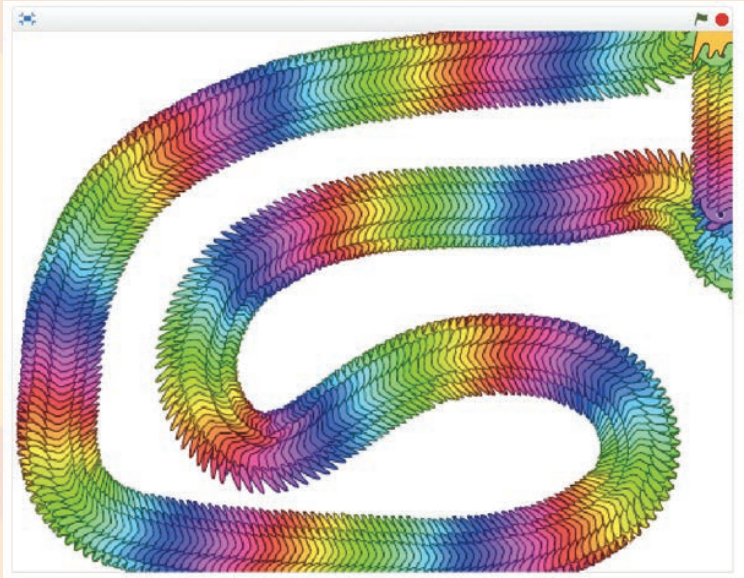
Şekil 16



Şekil 17

Bu kod bloğuna biraz renk katmak hiç zor olmayacaktır. Sürekli tekrarla bloğuna görünüm sekmesinin altında bulunan görsel efektlerden renk değiştirmeyi ekleyelim. (Şekil-18, 19)

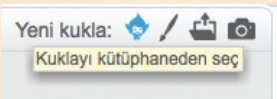
Şekil 18



Şekil 19

Gobo biraz gezdiğinde ekranı silmek istersek, kalem sekmesinin altında bulunan **Temizle** kod bloğuyla ekranı temizleyebiliriz. Bunu en basit şekilde; klavyedeki bir tuşa basarak yapabileceğimiz gibi, bu bölümde öğreneceğimiz şekilde ekleyeceğimiz bir düğme ile de yapabiliriz.

Öncelikle, Scratch kütüphanesinden bir düğme seçelim ve içeri aktaralım. (Şekil-20, 21)



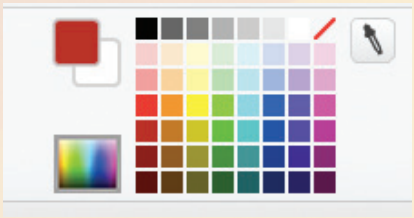
Şekil 20

Şekil 21

Düğmemizin üzerine bir yazı yazmak istiyoruz. Bunun için düğmemizin kılığı üzerinde bir değişiklik yapmalıyız. Orta sekmede bulunan kılık bölümünü seçelim. (Şekil-22)

Şekil 22

Ardından, kılık üzerinde yazı yazabilmek için ekranın sağında bulunan yazım aracını seçelim. Yazının rengini belirlemek için orta alanda bulunan renk seçim paletinden istediğimiz rengi seçebiliriz. (Şekil-23, 24)



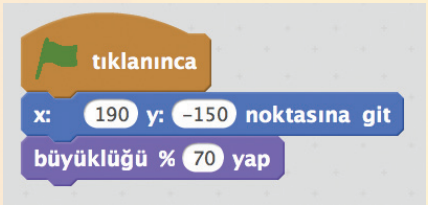
Şekil 23

Şekil 24

Düğmenin üzerine istediğimiz yazıyı yazalım. Dilersek, altta bulunan font sekmesinden istediğimiz fontu seçebiliriz. (Şekil-25)

Şekil 25

Şimdi sıra geldi kod bloğumuzu yazmaya. Öncelikle, düğmemizi biraz küçültüp ekranın bir köşesine gitmesini sağlayalım. (Şekil-26)



Şekil 26

Ardından, düğmeye tıklandıkça ekranın temizlenmesini sağlayalım (Şekil-27)

Şekil 27

Kalem sekmesiyle yapabileceklerimiz elbet bu kadar değil. Sadece kalem sekmesi kullanılarak yapılan oyunlara göz atın isterseniz. Bir sonraki kitapta belki biz de böyle oyunlardan birini anlatırız.

Sadece kalem kullanılarak yapılmış projelerden seçmeler:

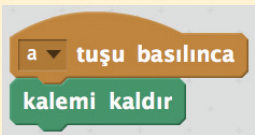
https://scratch.mit.edu/projects/75258684/

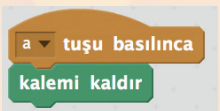
https://scratch.mit.edu/projects/72986156/

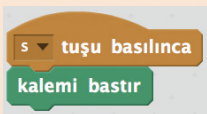
https://scratch.mit.edu/projects/79197216/

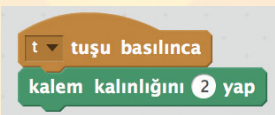
https://scratch.mit.edu/projects/52527806/

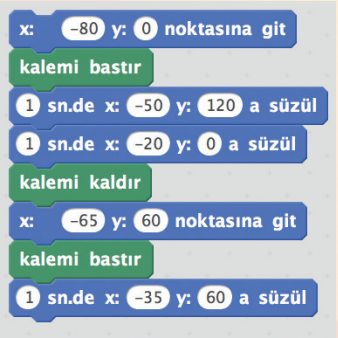
BU BÖLÜMDE KULLANILAN KOD BLOKLARI

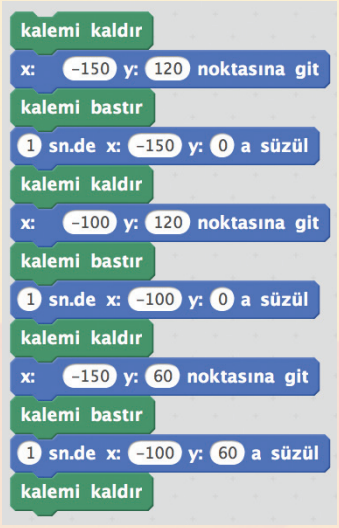








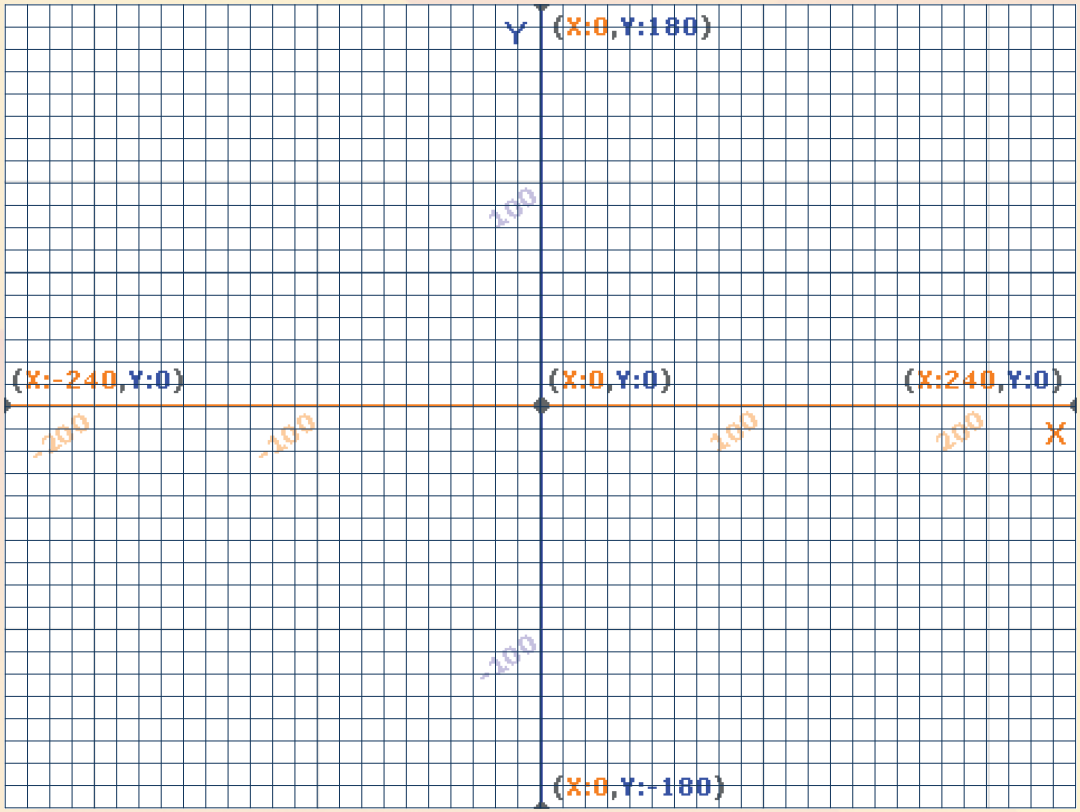




Scratch programlama dilini çok yakında tabletlerinizle de kullanabileceksiniz. Ayrıntılar için http://scratched.gse.harvard.edu/ adresini takip edebilirsiniz.

ÇALIŞMA SAYFASI

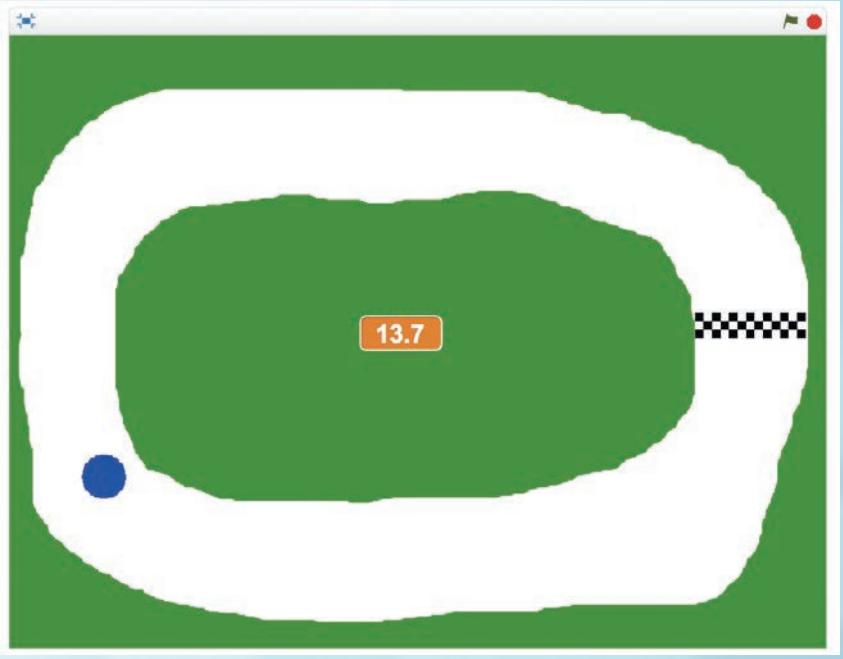
Uygulamada yaptığımız gibi kalem sekmesi ile kendi adınızı ya da istediğiniz bir kelimeyi yazınız. Yazmadan önce aşağıdaki çalışma kâğıdını kullanarak harfleri oluşturmak için gerekli çizgilerin başlangıç ve bitiş noktalarını belirleyebilirsiniz.



Bilye Yarışı

Bu Aktivitede Neler Öğreneceğiz?

- Yatay ve dikey hız kullanımını,
- İvme kavramını,
- Düzgün yavaşlamayı,
- Sayaç oluşturmayı.



BİLGİ

Algılama sekmesinin altında bulunan zamanlayıcı, yeşil bayrağa basar basmaz çalışmaya başlar ve bu zamanlayıcıyı istediğimiz zaman sıfırlayabiliriz.

EK GÖREV

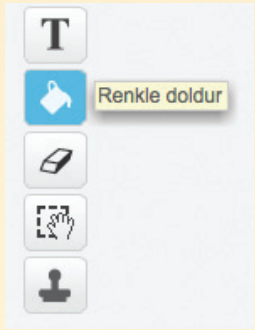
- Her turda pisti değiştirelim.
- Tur sayısı arttıkça aracı hızlandıralım.
- Tur atmak için zaman sınırlaması koyalım.

Bilye Yarışı

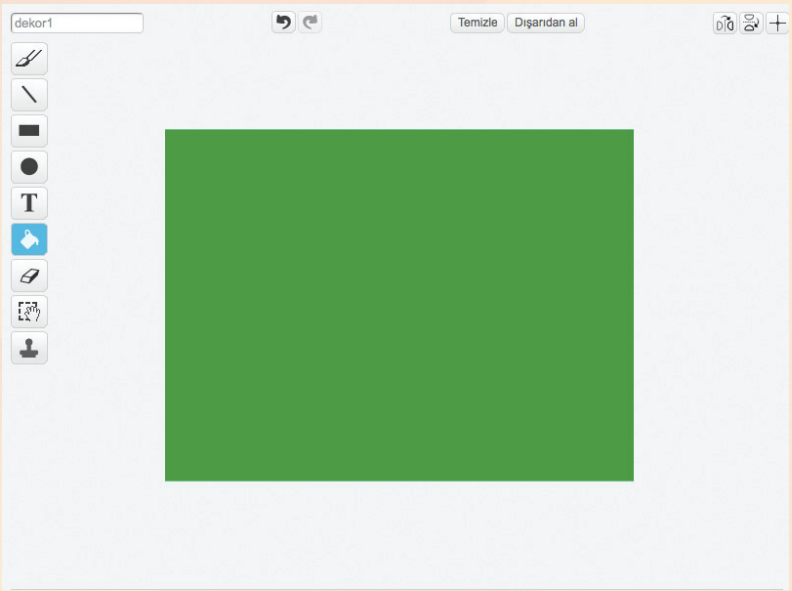
Küçük bir yarış oyunu hazırlayacağız. Pistin etrafında tur atacağız ve bir tur atma süremizi hesaplayacağız. (Şekil-1)

Şekil 1

Öncelikle, pistimizi oluşturalım. Sahne üzerine gelip Kılıklar sekmesini tıklayalım. Ve Renkle doldur düğmesine tıklayıp sahneyi yeşil renk ile dolduralım.
(Şekil-2, 3)



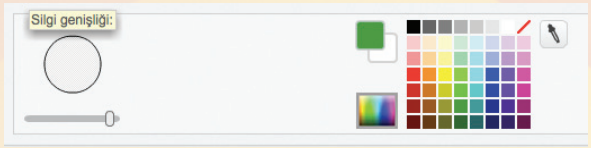
Şekil 2



Şekil 3

Silgimizi alıp istediğimiz boyuta getirerek pist için istediğimiz şekilde bir alan oluşturalım. (Şekil-4, 5, 6)

Şekil 4



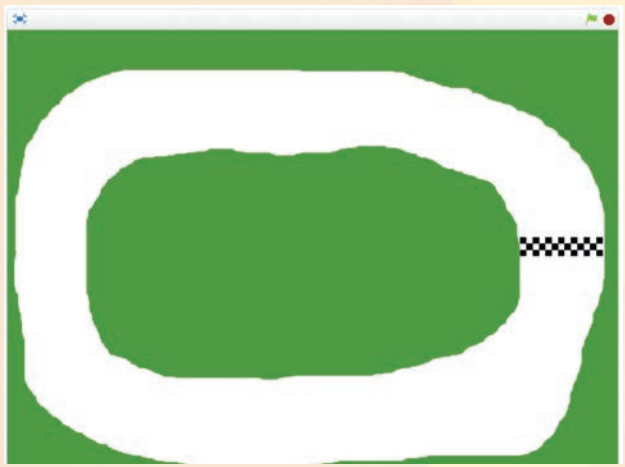
Şekil 5

Şekil 6

Ardından, **www.ogreniyorum.xyz** adresinden bitiş çizgisi kuklasını indirip bilgisayarımıza aktarabilir ya da kendi bitiş çizgimizi oluşturabiliriz. (Şekil-7)

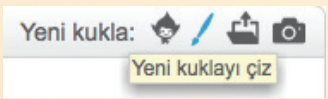
Şekil 7

Bitiş çizgisini istediğimiz yere getirelim. (Şekil-8)

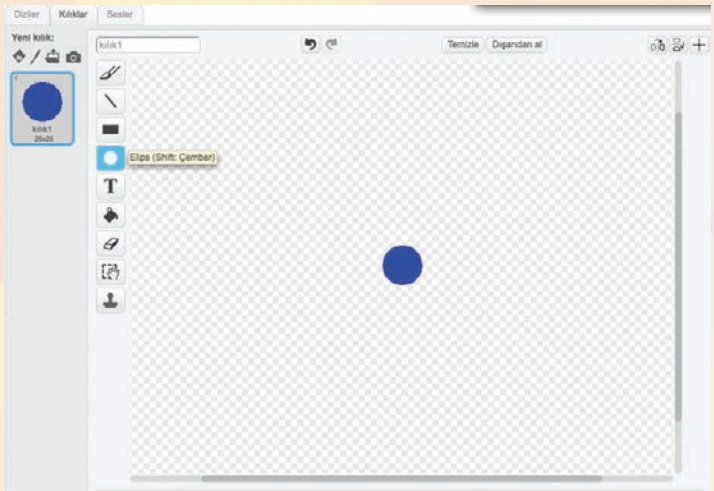


Şekil 8

Kendimize aracımız yerine geçecek bir kukla oluşturalım. Biz bir daire kullanacağız. Yeni kukla çiz komutundan bir daire oluşturalım.
(Şekil-9, 10)



Şekil 9



Şekil 10

Öncelikle; aracımızı başlamak istediğimiz noktaya götürelim ve oyun her başladığında burada olmasını sağlamak için, yeşil bayrak tıklandığında bu koordinata gitmesini sağlayalım. (Şekil-11)

Şekil 11

Aracımıza yön vermek için yön tuşlarını kullanacağız. Bu, diğer oyunlardaki hareketimizden biraz farklı bir şekilde olacak. **Dikey hız** ve **yatay hız** adında iki tane değişken oluşturalım. Başlangıçta bu iki değişken de 0 değerini alacak. (Şekil-12)

Şekil 12

Bilyemiz Düzgün Hızlanıyor

Dikey hızın, yukarı tuşa bastığımızda artmasını, aşağı tuşa bastığımızda azalmasını sağlayacağız. Yatay hızın; sağ oka bastığımızda artmasını, sol oka bastığımızda azalmasını sağlayacağız. (Şekil-13)

Şekil 13

Şu an oyunu çalıştırdığımızda yön tuşlarını kullanarak aracımızı hareket ettiremediğimizi görüyoruz. Çünkü, aracımıza koordinatları için herhangi bir komut vermedik. Yeşil bayrak tıklandığında aracımız sürekli olarak x ve y ekseninde yatay ve düşey hızı kadar ilerleyecek ya da gerileyecek. (Şekil-14)

Şekil 14

Aracımız, pistin içinde de dışında da aynı hızla ilerliyor. Pistin dışına çıktığımızda aracımız daha zor ilerlemeli. Aracımız pistin dışına çıktığında yani yeşil zemine değdiğinde yatay ve düşey hızı 0.95 sayısı ile çarparak düzgün bir şekilde yavaşlamasını sağlayalım. (Şekil-15)

Şekil 15

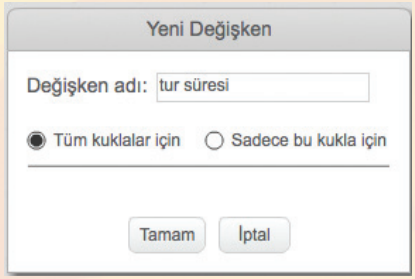
Aracımız, istediğimiz gibi hareket ediyor ve pistin etrafında tur atabiliyor. Bitiş çizgisine gelene kadar geçen süreyi hesaplamak için, **zaman** adında bir değişken oluşturalım. (Şekil-16)

Şekil 16

Yeşil bayrak tıklandığında, Zaman değişkenini sıfırlayalım. Zaman değişkeninin onda bir hassasiyetle olması bizim için yeterli. Bu yüzden her 0.1 saniyede bir zaman değişkenini 0.1 artıracağız. (Şekil-17)

Şekil 17

Öncelikle, **tur süresi** adında bir değişken oluşturalım. Bu değişkeni, aracımız her tur attığında geçen süreyi ekranda göstermek için kullanacağız. (Şekil-18)



Şekil 18

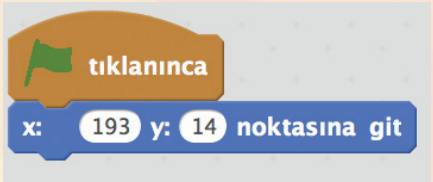
Sürekli olarak aracımızın bitiş çizgisine değip değmediğini kontrol edelim. Bitiş çizgisine değer değmez tur süresi değişkenini zaman değişkenine eşitleyelim. Hemen ardından yeni tur için zaman değişkenini sıfırlayalım. 2 saniye boyunca aracımız turu kaç saniyede bitirdiğini ekranda göstersin. (Şekil-19)

Şekil 19

BİLGİ

Sayaç oluşturmak için algılama sekmesinin altında bulunan süre ölçeri kullanabilirsiniz. Süre ölçer 1/10 hassasiyette, geçen süre bilgisini size verecektir.

BU BÖLÜMDE KULLANILAN KOD BLOKLARI



ÇALIŞMA SAYFASI

1. Aşağıdaki kod blokları çalıştıktan sonra değişkenlerin son değerlerini bulunuz.

.....

.....

.....

.....

.....

.....

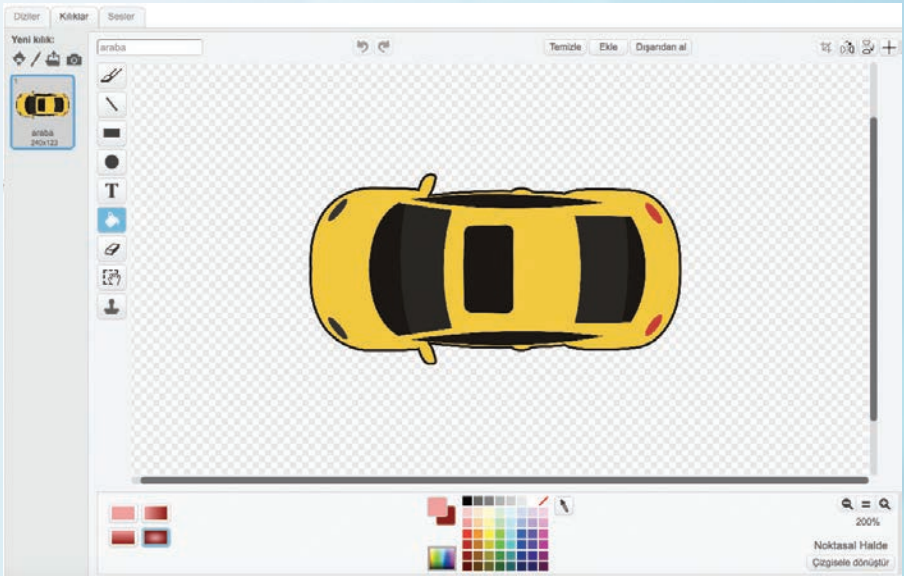
.....

.....

Karşıya Geçme

Bu Aktivitede Neler Öğreneceğiz?

- Klavye ile karakter kontrolünü,
- Görünüm sekmesi altındaki komutları,
- Koordinat düzlemini,
- Programlamada **veya** bağlacının kullanımını,
- Bir oyuna yaşam eklemeyi.



EK GÖREV

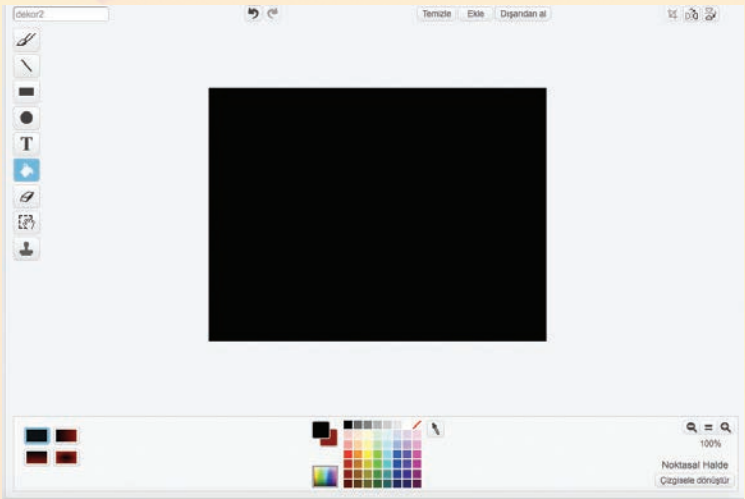
- Oyundaki can sayısını sembollerle göstererek, can kaybettikçe bu sembollerin eksilmesini sağlayalım.
- Oyun içerisine farklı modelde arabalar ekleyelim.

Yolumuzu Oluşturalım

Scratch programımızı açalım. Oyunumuza istediğimiz ismi verebilirsiniz, biz "**karşıya geçme**" olarak adlandıracağız. Kedimizi silip oyunumuza başlayalım. (Şekil-1)

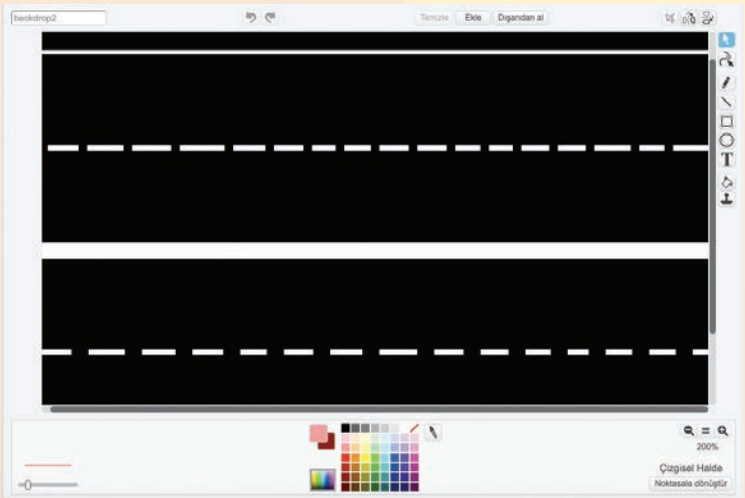
Şekil 1

Öncelikle, arka zeminimizi oluşturalım. İki tane çift şeritli yol düşünelim ve başlamak için arka zemini siyah yapalım. Dekorlar sekmesine gelip zemini siyaha boyayalım. (Şekil-2)



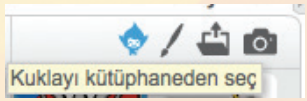
Şekil 2

Ardından, ekte bulunan (Şekil-3)'teki yol çizgilerini oluşturalım. Renkleri istediğimiz gibi değiştirebiliriz. Bu zemin resmini Scratch editörü kullanarak yapabileceğimiz gibi, paint ve benzeri editörlerle de yapabilir ya da hazır olarak ekteki dosyadan alabiliriz.



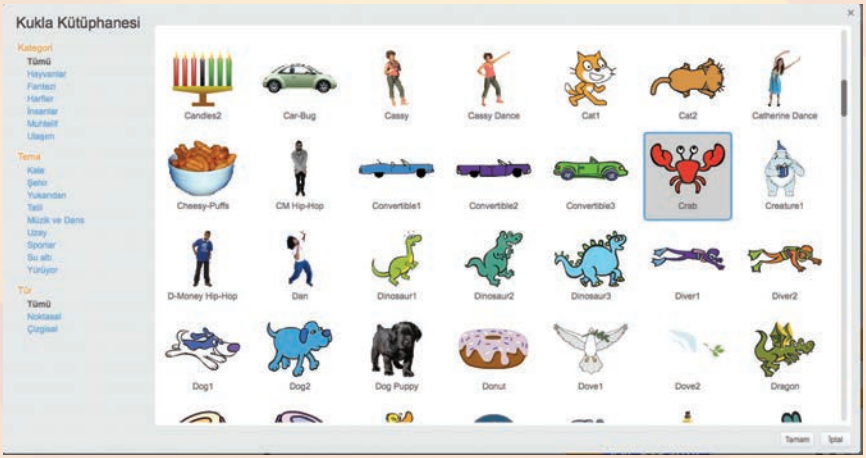
Şekil 3

Karşıya geçmesi için, istediğimiz herhangi bir nesneyi seçebiliriz. “Kuklayı kütüphaneden seç” düğmesinden kütüphane aracılığıyla bir kukla seçelim. (Şekil-4)



Şekil 4

Biz yengeç karakterini seçiyoruz. (Şekil-5)

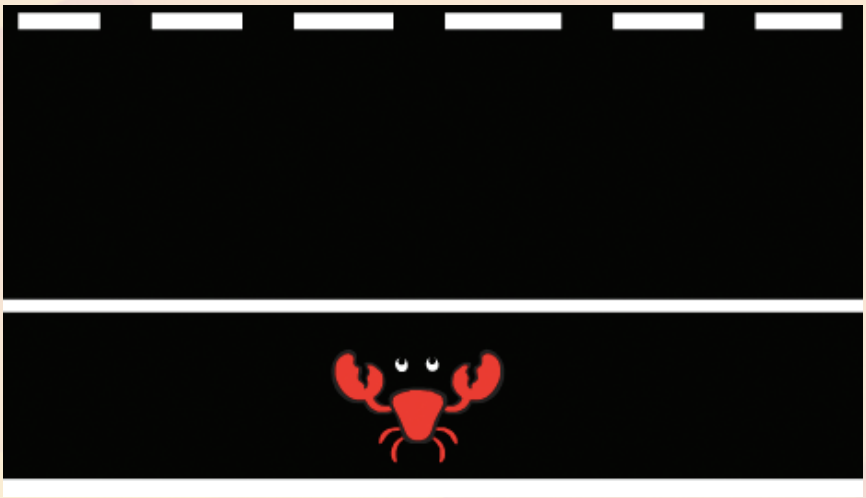


Şekil 5

Eklenen yengeç karakterini, ekranda istediğimiz boyuta getirmek için “küçült” düğmesine tıklayıp ardından yengecimizin üzerine tıklayabiliriz. (Şekil-6)

Şekil 6

Ve yengecimiz artık kodlama için hazır. (Şekil-7)



Şekil 7

Yengeç Hareket Ediyor

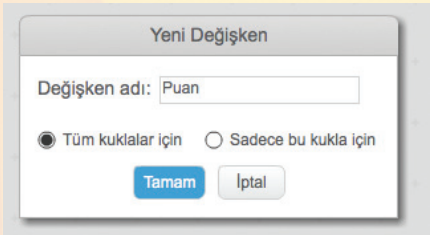
Programlamanın en temel öğelerinden biri, başlangıç değeri atamadır. Biz de bunu unutmadan, yeşil bayrak tıklandığında yengecimizin nerede durmasını istediğimize karar veriyoruz ve yeşil bayrak tıklanınca durması gereken koordinatı atıyoruz. (Şekil-8)

Şekil 8

Karakteri seçtikten sonra, klavye ile hareketini sağlayalım. Yengecimiz ileri, geri, sağ ve sol olmak üzere 4 yöne hareket edecek. İster yön tuşlarını; ister w, a, s, d tuşlarını hareket için atayabilirsiniz. Biz, hareketi yön tuşlarıyla sağlayacağız. (Şekil-9)

Şekil 9

Yengecimiz ile ilgili bir eklenti daha yapalım. Yengecimiz yolun karşısına geçtiğinde skor bir artmalı ve yengeç başlangıç noktasına dönmeli. Bunun için öncelikle bir Skor değişkeni tanımlayalım. (Şekil-10)



Şekil 10

Değişkenimizin üstüne tıklayarak görünümünü yalnızca sayı görünecek şekilde değiştirelim. (Şekil-11)

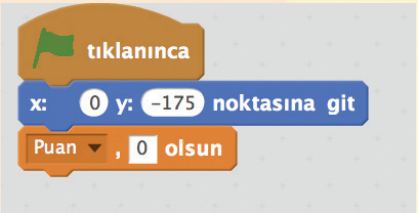
Şekil 11

Değişken oluşturur oluşturmaz ilk değerini atayalım. (Şekil-12)

Şekil 12

Ardından, karşıya her geçişte skoru bir artırıp yengeci başlangıç noktasına döndürecek bir kodlama yapalım. (Şekil-13)

Yengecimiz boş bir zemin üzerinde hareket edebiliyor ve karşıya geçince puanı artıyor. Şimdi oyunumuza araç ekleyelim. Araba resmini ek dosyalar arasında bulabilirsiniz. Bu resmi uygulamamızda açtıktan sonra, kodlarımızı yazmaya başlayalım. Arabaları iki farklı şekilde oluşturacağız. Birinci yöntemde bir arabayı yapıp ardından kopyalarını çıkartıp el ile düzenlemeler yapacağız, ikinci olarak klonlamayı kullanacağız.



Şekil 13

Öncelikle, araçlarımızdan birinin kodunu yazalım. Başlangıçta olmasını istediğimiz yeri ilk değer olarak atayalım. (Şekil-14)

Şekil 14

Yeşil bayrak tıklandığında arabamız başlangıç noktasına gitsin. (Şekil-15)

Şekil 15

Şimdi ayrı bir yeşil bayrak tıklanınca alalım ve aracımızın en sola kadar gidip tekrar başa dönmesini sağlayalım. Bunun için; sürekli olarak aracımızın x değerini artırmalı (sağa doğru gitmeli), ekranın en sağına geldiğinde yani x değeri 240'tan büyük olduğunda en sola gitmeli (x değeri -240 olmalı). Araba2, Araba3 ve Araba4 için sırasıyla (Şekil-17), (Şekil-18), (Şekil-19)'daki gibi y eksenini değiştirmek yeterli olacaktır.

Şekil 16

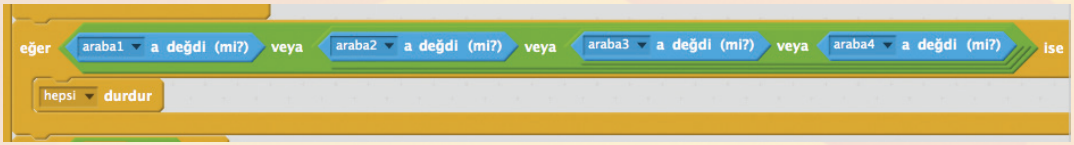
Şekil 17

Şekil 18

Şekil 19

Arabalarımız hep aynı anda, aynı hızla ve aynı yöne gidiyorlar. Bu problemleri birazdan gidereceğiz, fakat daha önce halletmemiz gereken bir sorun var: Yengecimiz arabalara çarpınca hiçbir şey olmuyor. Şimdilik tek canımız olduğunu ve herhangi bir arabaya değdiğimizde oyunun bittiğini düşünelim. O halde, yengece arabalardan herhangi biri çarptığında oyunun durmasını sağlayalım. Bunun için, işlem bloğunun altında bulunan "veya" bloğunu bolca kullanacağız.

Yengeç; Araba1, Araba2, Araba3 ya da Araba4'e değdiğinde programın hepsini durduracağız. (Şekil-20)



Şekil 20

Yengeç Karşıya Geçmeye Çalışıyor

Artık oyunumuz çalışıyor diyebiliriz. Şu an için güzel görünmeyen şeylerden biri, arabaların aynı şekilde hareket etmesi. Alt iki şeritteki arabalar soldan sağa gitmeye devam etsin ama rastgele bir süre bekleyerek, aynı zamanda yola çıkmalarını engelleyelim. Araba1 ve Araba2'nin kod sahnesinde (Şekil-21) ve (Şekil-22)'deki değişiklikleri yapalım. Böylece arabaların rastgele bir süre boyunca beklemesini ve aynı süreyle çıkmamasını sağlayabiliriz.

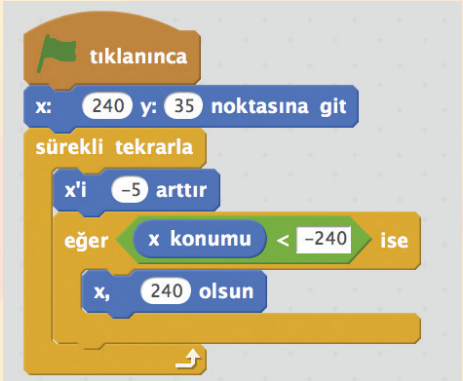
Şekil 21

Şekil 22

Araba 3 ve 4'ün ise soldan sağa değil, sağdan sola hareket etmelerini sağlamalıyız. Öncelikle, Araba 3 için kodda değişiklik yapalım. Aracımızın başlangıç noktasının x pozisyonu -240 değil, +240 olmalı. (Şekil-23)

Şekil 23

Aracımız soldan sağa değil, sağdan sola hareket etmeli. Bunun için x ekseninde sola gitmeli, yani -5 ilerlemeli. Ve x ekseninde -240'dan daha sola gittiğinde ekrandan çıkmış olduğunu anlayıp en sağa geri dönmeli. (Şekil-24)

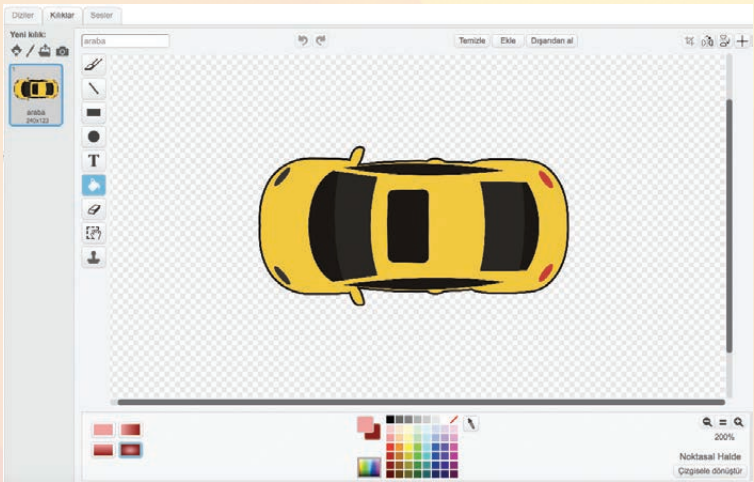


Şekil 24

Bu kodun benzerini Araba4 için de uyguluyoruz. (Şekil-25)

Şekil 25

Oyunumuzu başlattığımızda bir sorun olduğunu fark ettiniz mi? Evet. Araba3 ve Araba4 ters gidiyorlar. Şimdi bunu düzeltelim. "Kılıklar" sekmesinin altında sağ üstte bulunan kısımdan sağı sola çevir düğmesine tıklayalım. (Şekil-26)



Şekil 26

Aynı şeyi Araba4 için de yapalım.

Dilerseniz, Araba3 ve Araba4'ün aynı zamanda çıkmalarını engellemek için rastgele süre bekleyeceği blokları ekleyebilirsiniz. (Şekil-27)

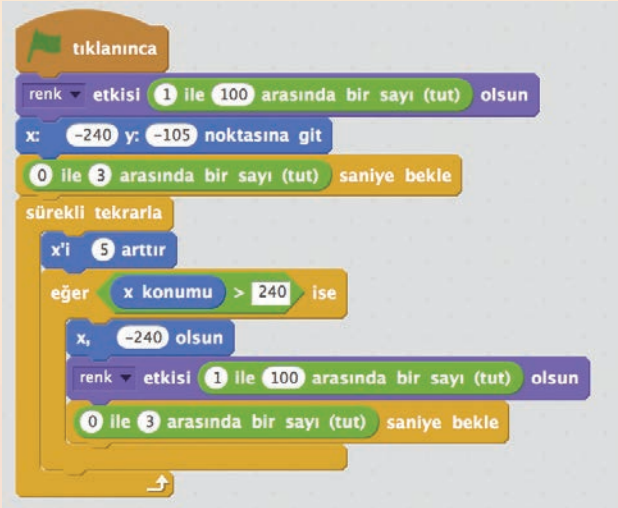
Şekil 27

Oyunumuz oynanmaya hazır, fakat birkaç değişiklikle çok daha güzel hale gelebilir. Örneğin, karakterimize birden fazla can verebiliriz. Oyun sırasında çıkacak bonuslarla puanımızı artırabileceğimiz gibi, oyundaki araçların hızlarının artıp azalmasını sağlayabiliriz ya da araçlarımızın renklerini değiştirip oyuna canlılık katabiliriz.

İlk önce, en basit değişiklik olan arabaların renk değişikliğini ayarlayalım. Araba1 için değiştireceğimiz bu kodu daha sonra tüm araçlar için değiştirmemiz gerekecek. Arabalar hareketine başlamadan önce, renk efekti ve rastgele bir değerle (0 ile 100 arası) renk değişimini sağlayalım.

Diğer tüm araçlara da bu kodu ekleyelim. Bu haliyle çalıştırdığımızda başlangıçta hepsi aynı renk olarak başlayacaklardır. Çünkü araçlara en sağa ya da sola geçtikten sonra yani en azından bir kez hareketi bitirince renk değiştirmesini söyledik. Başlangıçta da aynı renk olmamalarını istiyorsak, döngünün içerisine eklediğimiz satırı başlangıca da eklemeliyiz. (Şekil-28-29)

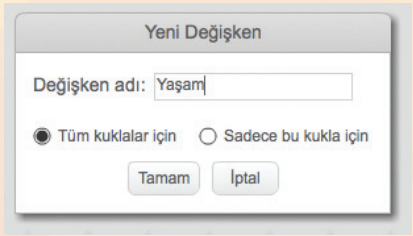
Şekil 28



Şekil 29

Yaşam Ekleyelim

Bu kod bloğunun en son halindeki değişikliği tüm arabalara uygulayalım. Artık tüm arabalarımız farklı renkte hareket ediyorlar. Şimdi yengecimize 3 can verelim. Öncelikle, kaç yaşamımız kaldığını anlayabilmemiz ve yaşamımız sıfır olduğunda oyunun bitmesi için "yaşam" adlı bir değişken tanımlayalım. (Şekil-30)

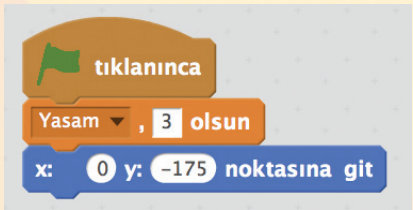


Şekil 30

Yaşam değişkeninin ekranda görünmesini istemiyoruz, o yüzden veri bloğunda Yaşam değişkeninin yanındaki kutudaki tiki kaldırıyoruz. (Şekil-31)

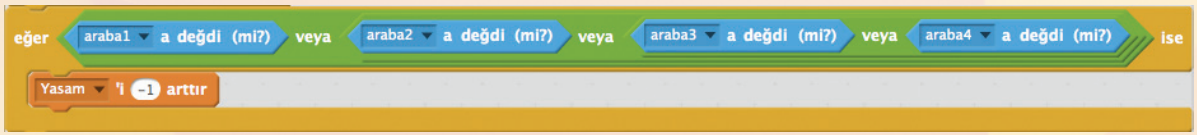
Şekil 31

Bir değişken tanımladığımızda yapmamız gereken ilk şey, bir ilk değeri varsa onu atamaktır. Siz istediğiniz kadar can atayabilirsiniz. Biz 3 can vereceğiz. O halde yaşam değişkenini yeşil bayrak tıklandığında 3 olacak şekilde atayalım. (Şekil-32)



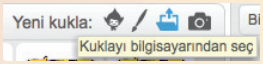
Şekil 32

Arabaya her çarpılışında bir yaşam eksilmeli ve sürekli olarak yaşamın sıfır olup olmadığını kontrol edilmeli. Yengeç karakterine gelip, arabaya çarpıp çarpmadığını kontrol ettiğimiz bölümde hepsini durdur komutunun olduğu yere yaşamı bir eksilt komutu tanımlayalım. (Şekil-33)

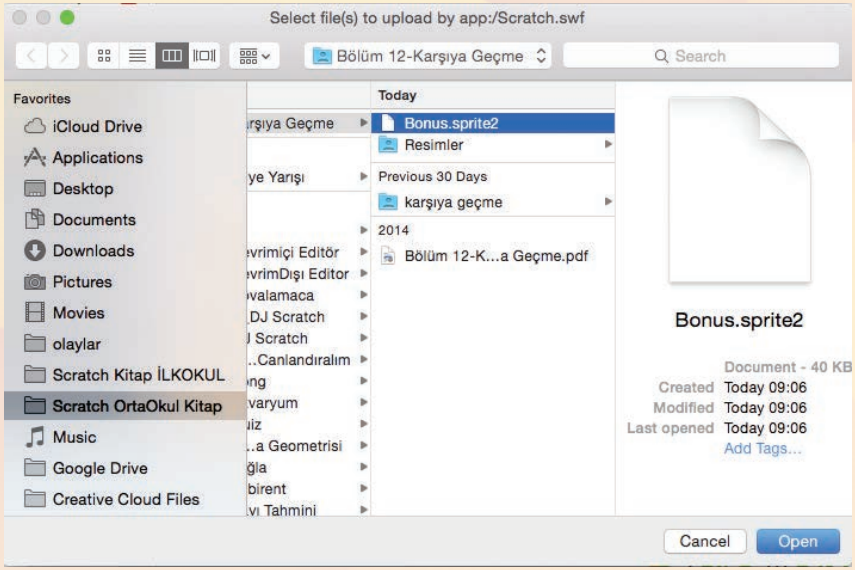


Şekil 33

Oyun içinde, oynanabilirlikle ilgili sorunları çözdük. Oyunu belli bir süre oynayıp özellikle hız, araba sayısı gibi şeyleri değiştirerek oyunu istediğiniz hıza getirebilirsiniz. Şimdi oyuna ek bir özellik katalım. Rastgele bir yerde çıkan bir mücevher bize ek 3 puan versin. Öncelikle, mücevher resmini dosyanın ekinden içeriye aktaralım. (Şekil-34, 35)



Şekil 34



Şekil 35

Bu karakterimizin adını bonus diye düzeltelim. (Şekil-36) Mücevherimizi, 3-10 saniye arasında rastgele bir sürede ortaya çıkıp 2 saniye içinde tekrar görünmez hale gelecek şekilde kodlayacağız.

Şekil 36

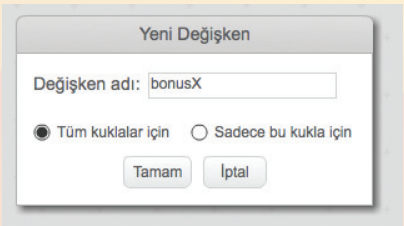
Öncelikle, yeşil bayrağa tıkladığımızda ekranda herhangi bir bonus görünmesini istemiyoruz. Bonus karakterine tıklayıp kod sahnesinde düzenleme yapalım. (Şekil-37)

Şekil 37

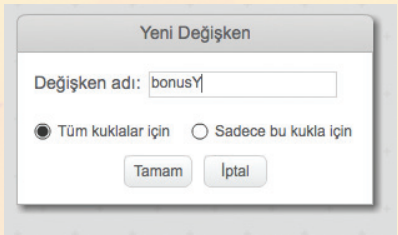
Mücevher, oyun devam ettiği sürece belirli bir zaman boyunca bekledikten sonra rastgele x ve y koordinatlarında ortaya çıksın. Öncelikle, bu kodun yalnızca yaşam sıfırdan büyük olduğu sürece çalışmasını sağlayalım. (Şekil-38)

Şekil 38

Ardından, rastgele x ve y koordinatlarımızı ayarlayalım. "Bonus X" ve "Bonus Y" adlı iki değişken oluşturalım. (Şekil-39, 40)

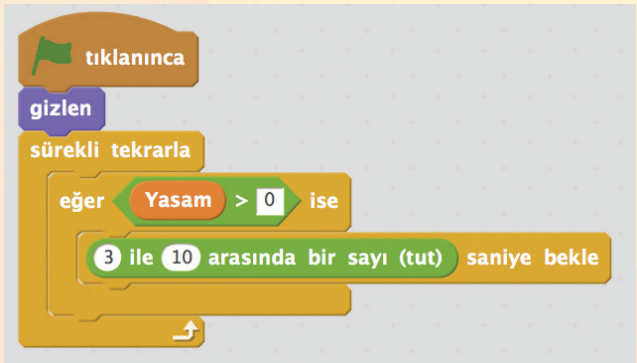


Şekil 39



Şekil 40

(Şekil-41)'de rastgele bir süre bekliyoruz. Ardından, x ve y koordinatlarını yine rastgele olacak şekilde belirtip ekranda görünmelerini sağlıyoruz. (Şekil-42)



Şekil 41

Şekil 42

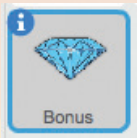
Karakterimiz (yengeç) bonusları aldıkça 3 puan (siz dilerseniz farklı bir puan da verebilirsiniz) alacak. Ve belirli bir saniye sonra mücevher, ekranın başka bir noktasında görünecek. Mücevher karakterinin kod sahnesine gidelim. Yengeç mücevheri aldıktan sonra mücevher yok olacak ve puanımız 3 artacaktır. (Şekil-43)

Şekil 43

BU BÖLÜMDE KULLANILAN KOD BLOKLARI

araba3

araba4




```
tıklanınca
gizlen
sürekli tekrarla
  eğer [Yasam] > [0] ise
    [3] ile [10] arasında bir sayı (tut) saniye bekle
    [bonusX] , [-200] ile [200] arasında bir sayı (tut) olsun
    [bonusY] , [-120] ile [120] arasında bir sayı (tut) olsun
    x: [bonusX] y: [bonusY] noktasına git
    görün
```

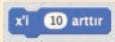
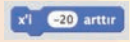
```
tıklanınca
sürekli tekrarla
  eğer [yengec] a değdi (mi?) ise
    [Puan] 'i [3] arttır
    gizlen
```

```
tıklanınca
[Puan] , [0] olsun
[Yasam] , [3] olsun
x: [0] y: [-175] noktasına git
sürekli tekrarla
  eğer [sağ ok] tuşu basılı (mı?) ise
    x'i [5] arttır
  eğer [sol ok] tuşu basılı (mı?) ise
    x'i [-5] arttır
```

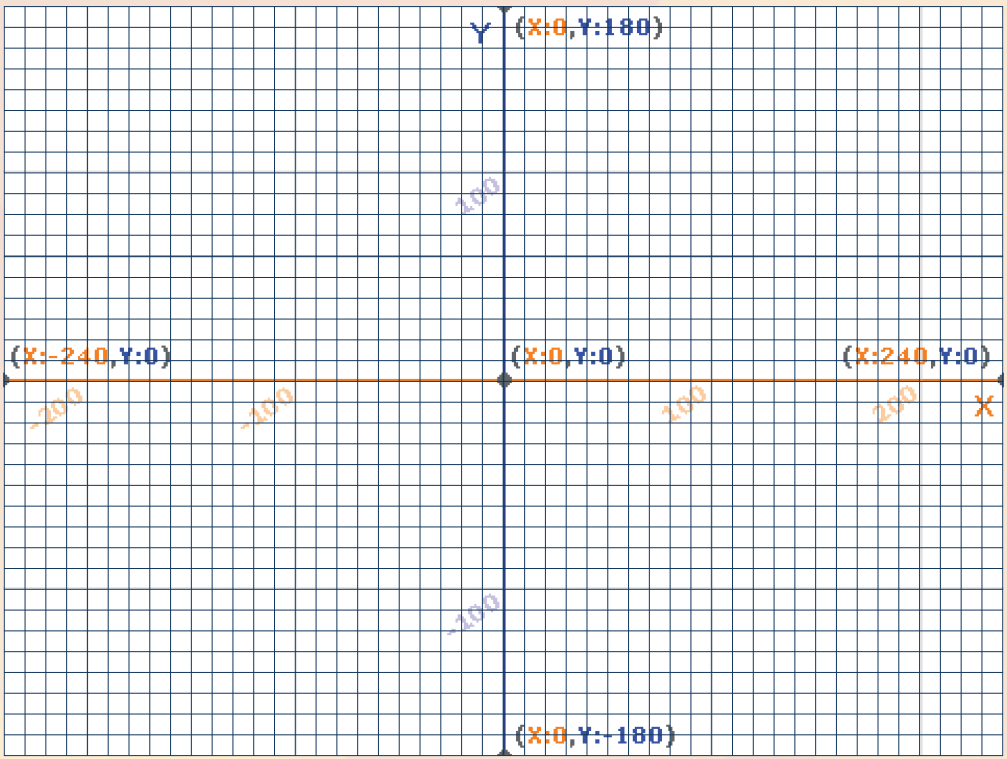
```
eğer [yukarı ok] tuşu basılı (mı?) ise
  y'yi [5] arttır
eğer [aşağı ok] tuşu basılı (mı?) ise
  y'yi [-5] arttır
eğer [y konumu] > [135] ise
  x: [0] y: [-175] noktasına git
  [Puan] 'i [1] arttır
```

```
eğer [araba1] a değdi (mi?) veya [araba2] a değdi (mi?) veya [araba3] a değdi (mi?) veya [araba4] a değdi (mi?) ise
  [10] defa tekrarla
    [hayalet] etkisini [10] arttır
  görsel etkileri temizle
  [Yasam] 'i [-1] arttır
  x: [0] y: [-175] noktasına git
eğer [Yasam] = [0] ise
  [oyunbitti] haberini sal ve bekle
```

ÇALIŞMA SAYFASI

X ekseni sağa sola gitmek demektir. Örneğin;  sağa 10 piksel git.  sola 20 piksel git anlamına gelir. Y ekseni yukarı aşağı gitmek demektir; aşağıya 10 piksel gitmek, yukarıya 20 piksel gitmek anlamına gelir.

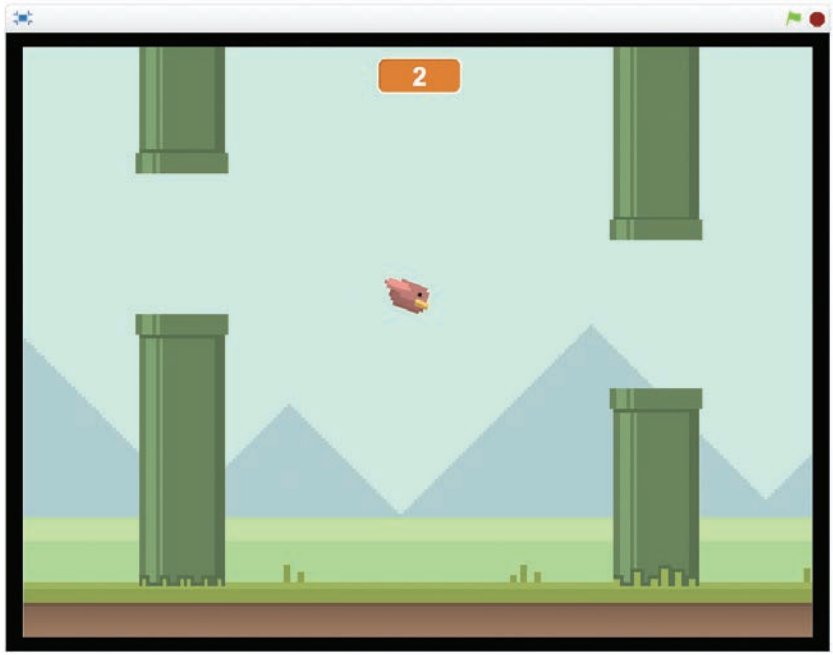
1. Yukarıdaki verilen şekle göre aşağıdaki boşlukları doldurunuz. (Her boşluk yengecin şu an bulunduğu duruma göre doldurulacaktır.)
   - **a.** Yengecimizin (-20,-60) noktasına gelmesi için ..... sağa ve ..... aşağıya gitmesi gerekmektedir.
   - **b.** Yengecimizin (-20,-60) noktasına gelmesi için ..... sağa ve ..... aşağıya gitmesi gerekmektedir.
   - **c.** Yengecimizin (100,40) noktasına gelmesi için ..... sağa ve ..... yukarı gitmesi gerekmektedir.
   - **d.** Yengecimizin (60,0) noktasına gelmesi için ..... sağa ve ..... aşağıya gitmesi gerekmektedir.
   - **e.** Yengecimizin (0,-50) noktasına gelmesi için ..... sağa ve ..... aşağıya gitmesi gerekmektedir.
   - **f.** Yengecimizin (0,140) noktasına gelmesi için ..... sağa ve ..... yukarı gitmesi gerekmektedir.
   - **g.** Yengecimizin (-50,40) noktasına gelmesi için ..... sağa ve ..... yukarı gitmesi gerekmektedir.



Zıplayan Kuş

Bu Aktivitede Neler Öğreneceğiz?

- Platform oyunu yapmayı,
- İkiz kullanmayı,
- Yerçekimi uygulamayı,
- Algılama sekmesini kullanmayı,
- Mantıksal operatörleri kullanmayı.



BİLGİ

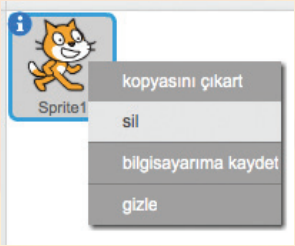
Scratch hakkında daha fazla kaynak arıyorsanız **http:// scratched.gse.harvard.edu** adresini ziyaret edebilirsiniz.

EK GÖREV

- Oyuna seviye ekleyelim.
- Flappy Bird karakterini değiştirelim.
- Çalıları ve engelleri değiştirelim.

Oyunumuzda kullanacağımız grafik ve dosyaları **www.ogreniyorum.xyz** adresinden indirebilirsiniz. Bu uygulamamızda, popüler bir oyun olan Flappy Bird oyununu kodlayacağız.

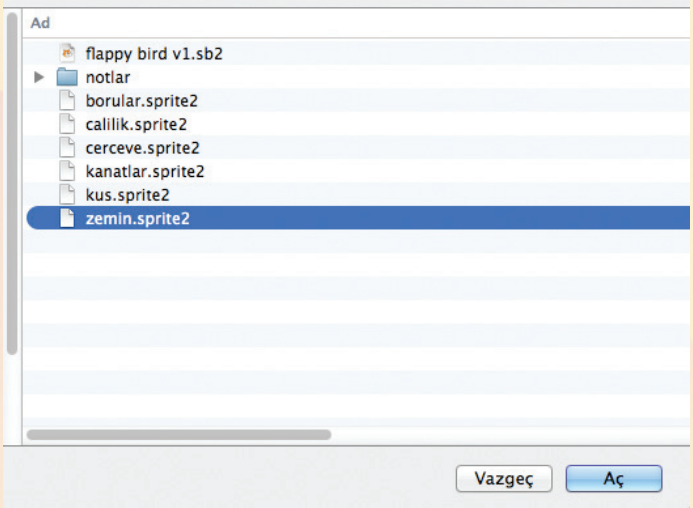
Kedi karakterimizi silerek oyunumuza başlayalım. (Şekil-1)



Şekil 1

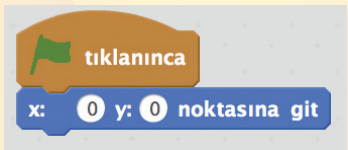
İlk olarak zeminimizi içeri aktaralım. Ekteki **zemin** karakterini içeri aktarın. (Şekil-2, 3)

Şekil 2



Şekil 3

Zemin karakterimizi, oyun her başladığında sağa sola kaymaması için merkeze alalım. (Şekil-4)

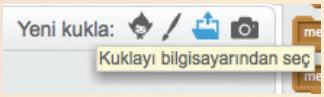


Şekil 4

Dilerseniz arka planı değiştirebilir ya da ileride oyuna seviye ekleyip farklı arka planlar kullanabilirsiniz.

Çalılık Ekleme ve Hareket

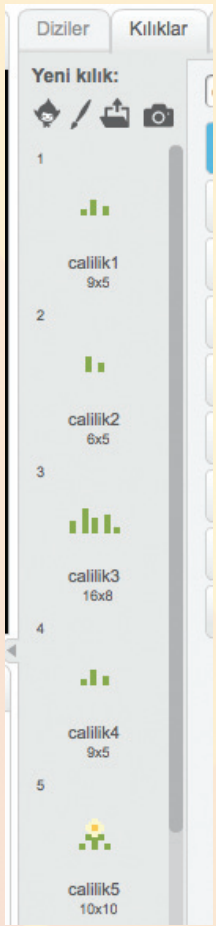
Çalılıklarımızı ekleyelim ve ekranı hareket edermiş gibi gösterelim. Öncelikle içeriye **çalılık** karakterini alalım. (Şekil-5, 6)



Şekil 5

Şekil 6

Çalılık karakterinin kılıklar sekmesine baktığınızda, 10 ayrı çalılık veya çiçek resmi göreceksiniz. Bunların ekranda rastgele ortaya çıkmasını ve zemin üzerinde geriye gitmesini sağlayarak sanki biz ileri gidiyormuşuz gibi hissetmemizi sağlayacağız. (Şekil-7)



Şekil 7

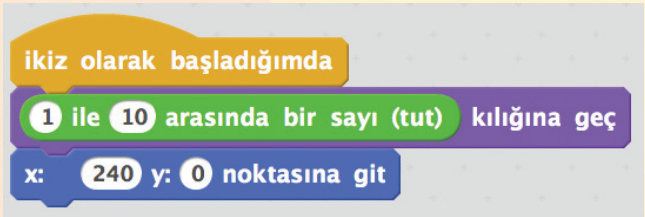
Öncelikle, çalılıklarımızı merkez noktasına getirelim. Çalılarımızı iki katı büyüklüğüne çıkarıp (%200) gizleyelim. Ortaya çıkacak çalılıklar bu çalı değil, bu çalıdan oluşturacağımız ikizler olacak. (Şekil-8)

Şekil 8

Belirli zaman aralıklarında bir tane çalılık oluşturalım. Bunun için, sürekli olarak rastgele bir saniye bekleyip bir ikiz oluşturmasını sağlamalıyız. (Şekil-9)

Şekil 9

Her ikiz oluştuğunda, aynı çalılık ya da çiçek gelmesin diye, 1 ile 10 arasında rastgele bir kılığa geçmesini sağlayalım. Ve ekranın en sağından başlaması için, (240, 0) koordinatını gönderelim. (Şekil-10)



Şekil 10

+ BİLGİ

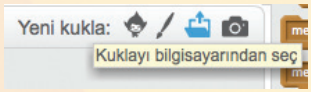
Shift tuşuna basılı tutarak yeşil bayrağa tıklarsak, Scratch turbo modda çalışır.

Oluşan ikizimiz ekranın en sağında, fakat görünür değil. Öncelikle görünüm sekmesinden, **görün** komutuyla ikizimizi görünür hale getirip ekranın en soluna gelene kadar (x konumu -235 değerinden küçük olana kadar) hareket etmesini sağlayalım. Oluşan ikizimizin hareketini x konumunu sürekli olarak azaltarak (burada 3'er 3'er azaltacağız) sağlayacağız. Ekranın dışına çıktığı anda, ikizimizi sileceğiz.
(Şekil-11)

Şekil 11

Engel Ekleme

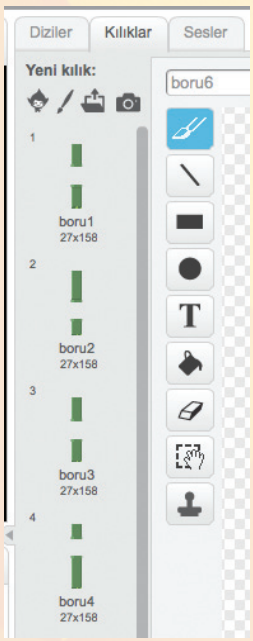
Çalılık eklemeye benzer bir şekilde engel ekleyeceğiz. Öncelikle, engeller karakterini **kuklayı bilgisayardan seç** komutuyla programımızın içine aktaralım. (Şekil-12)



Şekil 12

Engeller karakterinin kılıklar sekmesine baktığınızda, 8 ayrı engel çeşidi göreceksiniz. Bir önceki çalılık karakterindeki gibi ikizler oluşturup ekranda sağdan sola gitmesini sağlayacağız. Öncelikle karakterimizin büyüklüğünü iki katına (%200) çıkaralım. Karakterimizi ortalamak için, (0,0) noktasına gönderip gizleyelim.
(Şekil-13, 14)

Şekil 13

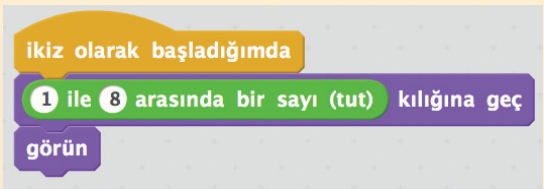


Şekil 14

Engellerin ortaya çıkışı, çalılar kadar sık olmamalı. 3 saniyede bir ikiz oluşturup ortaya bir engel çıkmasını sağlayalım. (Şekil-15)

Şekil 15

Çalılıklarda olduğu gibi, her ikiz oluştuğunda kılıklardan rastgele birini seçip görünür yapacağız. (Şekil-16)



Şekil 16

Karakterimizi ekranın en sağına alıp (240, 0) ekranın en soluna gelene kadar, sola doğru x konumunu 3 azaltacağız. Engel ekranın en soluna geldiğinde ise, döngüden çıkıp kendisini silecek. (Şekil-17, 18)

Şekil 17

Şekil 18

Kuş ve Hareketi

Ekte bulunan kuş karakterini programımızın içine aktaralım. (Şekil-19)

Şekil 19

Yeşil bayrak tıklanınca, kuşun daima ilk kılık görünümde olmasını ve merkeze geçmesini sağlayacağız. Ayrıca, kuş başlangıçta yatay pozisyonda olacağı için sağa dönmesini sağlayalım. (Şekil-20)

Şekil 20

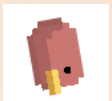
Bu oyunda, aslında kuş hiçbir zaman yatay bir hareket yapmayacak. Sadece, yerçekiminin ya da klavyeden bir tuşun etkisi ile dikey hareketler yapacak. Şu an oyunu çalıştırdığınızda, kuşun orta noktada süzüldüğünü göreceksiniz.

Kuş, yerçekiminin etkisiyle sürekli olarak aşağı düşmeli. Bunun gerçekçi olması için, kuş belirli bir açıyla aşağıya doğru dönmeli. (Şekil-21a, 21b, 21c, 21d, 21e)

Şekil 21a

Şekil 21b

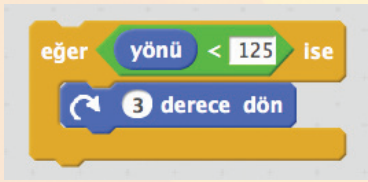
Şekil 21c



Şekil 21d

Şekil 21e

Şekilde gördüğünüz gibi, sürekli dönmeye devam ederse ters dönmüş olacak ki; bu istemediğimiz bir durum. Bunun için, açı belli bir dereceye gelinceye kadar dönmeli, sonrasında daha fazla dönmemeli. (Şekil-22)



Şekil 22

Yeşil bayrak tıklanınca, çalılıkların ve engellerin sola gitmesi sayesinde, sanki kuş karakteri sağa doğru gidiyormuş gibi görünecek. Biz de, yeşil bayrak tıklanınca kuşun aşağıya doğru düşmesini ve aşağıya dönmesini sağlayalım. (Şekil-23)

Şekil 23

Peki kuş karakterimiz nasıl uçacak, yani yukarıya çıkmasını nasıl sağlayacağız? Klavyeden herhangi bir tuşu, bunu yapmak için programlamalıyız. Biz boşluk tuşuna bastıkça, kuş karakterimizin bir müddet yukarı doğru çıkmasını sağlayacağız. Bunun için, yeşil bayrak tıklanınca sürekli olarak boşluk tuşuna basılıp basılmadığını kontrol edelim. Boşluk tuşuna basıldığında, **uç** diye bir haber gönderelim. (Şekil-24)

Şekil 24

Uç haberi gelince, yerçekiminin etkisiyle yaptığı hareketin tam tersini belirli bir miktar yapmalı. Biz on kez tekrar etmesini sağlayacağız, bu sayıyı dilerseniz değiştirebilirsiniz. Sola doğru 3'er derece dönmesini ve y ekseninde 3 birim yukarı çıkmasını sağlayacağız. (Şekil-25)

Şekil 25

Peki bu oyun nasıl bitecek? Kuş karakterimiz engellerden birine çarpınca ya da zemine değince oyun bitmeli. Bunun için, yeşil bayrak tıklandığında kuş karakterimiz engellerden birine çarpana ya da **y konumu** -140 değerinden küçük olana (yani yere çarpana) kadar beklemeliyiz. Eğer bu olaylardan birisi olursa, **oyun bitti** haberi göndermeliyiz. (Şekil-26)

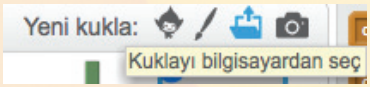
Şekil 26

Oyun bitti haberi geldiğinde, ufak bir efekt ekleyerek oyunu bitirelim. 4 defa tekrarlanacak bir döngü ile, kuş karakterimiz kılık değiştirecek ve hayalet efekti ile her karakter değişiminde biraz daha şeffaf bir karakterimiz olacak. Ardından, oyunumuzu tamamen durduracağız. (Şekil-27)

Şekil 27

Kanatlar

Şimdi kuş karakterimize kanatlarını takalım. Öncelikle, kanat karakterini oyunumuzun içine aktaralım. (Şekil-28)



Şekil 28

Yeşil bayrak tıklandığında, kanatların kuşun bulunduğu yönde olmasını ve kuş ile yapışık olmasını sağlayalım. (Şekil-29)

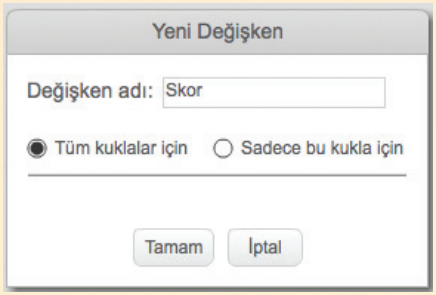
Şekil 29

Oyun bittiğinde kanatların ortalıkta görünmemesi için, oyun bitti haberi aldığında kuş karakterine yaptığımız gibi, hayalet etkisi ile karakterimizi görünmez yapalım. (Şekil-30)

Şekil 30

Skor Eklemeyi Unuttuk Sanmayın

Oyunda skor yok! Unuttuk sanmayın, en son olarak skor ekleyip oyunumuzu bitireceğiz. Öncelikle, **skor** adlı bir değişken oluşturalım. (Şekil-31)

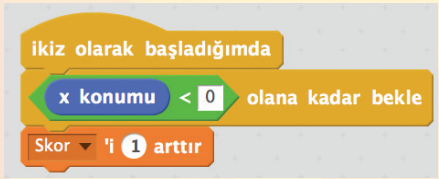


Şekil 31

Engeller karakterine gelip yeşil bayrak tıklanınca, Skor değişkeninin 0 değerini almasını sağlayalım. (Şekil-32)

Şekil 32

Her engel ikiz olarak başlatıldığında eğer kuş karakterini geçmişse yani apsisi (x değeri) 0 olmuşsa, Skor'u bir artıralım. (Şekil-33)

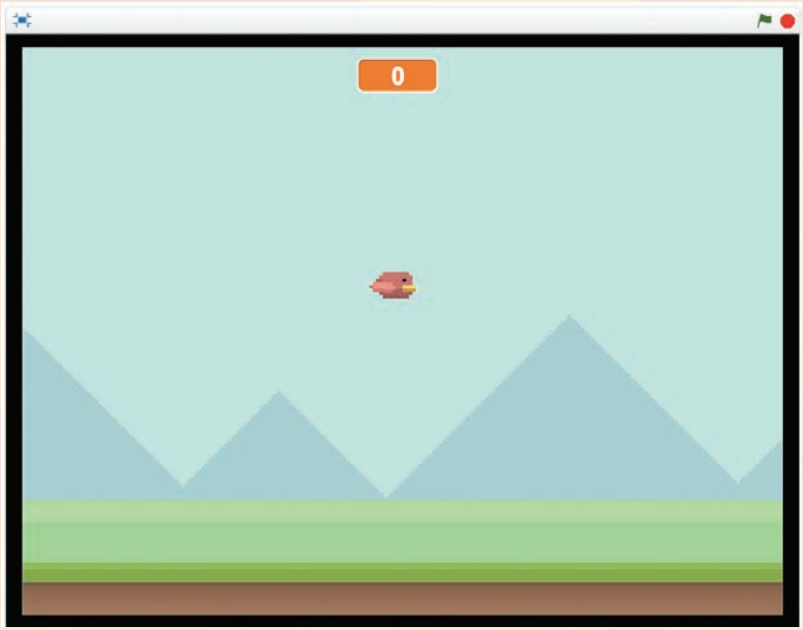


Şekil 33

Skor değişkeninin daha büyük görünmesi için, üzerine sağ klik yapıp büyük görünümü seçelim. (Şekil-34)

Şekil 34

Son olarak, skor değişkenini ortalayalım ve oyunumuz oynanmaya hazır. (Şekil-35)



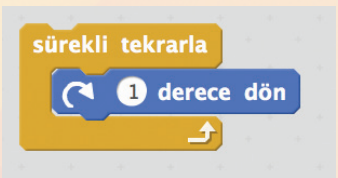
Şekil 35

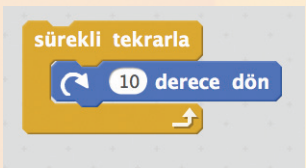
ÇALIŞMA SAYFASI

1. Scratch üzerinde, 1 saniyede 30 kare oynamaktadır. Buna göre, aşağıdaki kod blokları ile ekranın belirlenen yerinden başlayan bir karakter en sağına geçmek için kaç saniye harcar?

2. Aşağıdaki kod blokları çalıştığında, kedinin son yönünü tahmin ediniz.

3. Aşağıdaki kod bloklarına göre, bir karakterin kendi etrafında tam tur atması kaç saniye sürecektir?

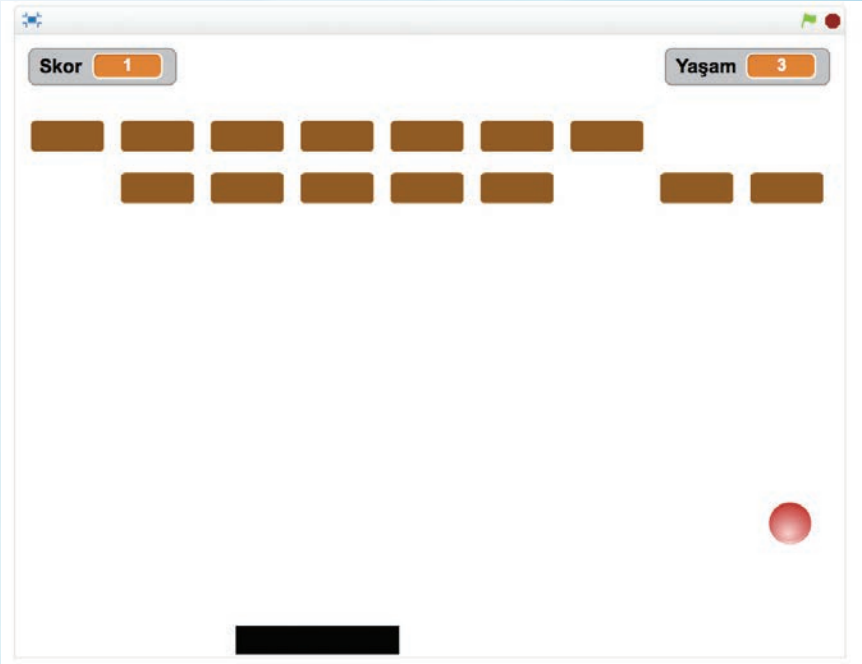




Tuğla Oyunu

Bu Aktivitede Neler Öğreneceğiz?

- Bir oyuna yaşam eklemeyi,
- İkiz oluşturmayı,
- Bir oyuna seviye eklemeyi,
- Haber yayımlamayı.



BİLGİ

Klon (ikiz) oluşturduğunuz uygulamalarda işiniz bittiğinde, klonu (ikizi) silmeyi unutmayın.

EK GÖREV

- Oyuna daha fazla seviye ekleyelim.
- Bazı tuğlaların, iki kere vurulunca yok olmasını sağlayalım.
- Oyuna çeşitli bonuslar ekleyelim.

Pong Oyununa Yaşam Ekleyelim

Bu oyun, daha önce yaptığımız PONG oyunu üzerine yapılacaktır. Eğer pong oyununu yapmamışsanız, o bölümü uygulamanız faydalı olacaktır.

Bu oyunda, pong oyunumuza yaşam ekleyeceğiz. Ayrıca, sahnenin üst kısmına tuğlalar ve oyuna seviyeler ilave edeceğiz.

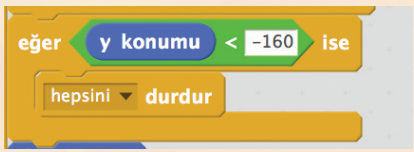
Öncelikle, oyunumuza yaşam ekleyelim. Veri sekmesinin altında, **yaşam** adlı bir değişken oluşturalım. (Şekil-1)

Şekil 1

Başlangıçta kaç yaşam olmasını istiyorsanız, yeşil bayrak tıklanınca **yaşam** değişkenine o değeri atayın. Top değişkenine gelip yeşil bayrak tıklanınca, **yaşam 3** olsun komutunu verelim. (Şekil-2)

Şekil 2

Pong oyununda, top zemine değdiğinde oyunu durdurmuştuk. Şimdi ise yaşamımızdan bir eksiltip topu başlangıç noktasına göndereceğiz. Top karakterinin üzerindeyken, komut satırında **hepsini durdur** komutunu silip yerine yaşamı -1 arttır, yani 1 eksilt komutunu yerleştireceğiz. Yaşamı 1 eksilttikten sonra, top tekrar başlangıç noktasına dönmeli. (Şekil-3, 4)



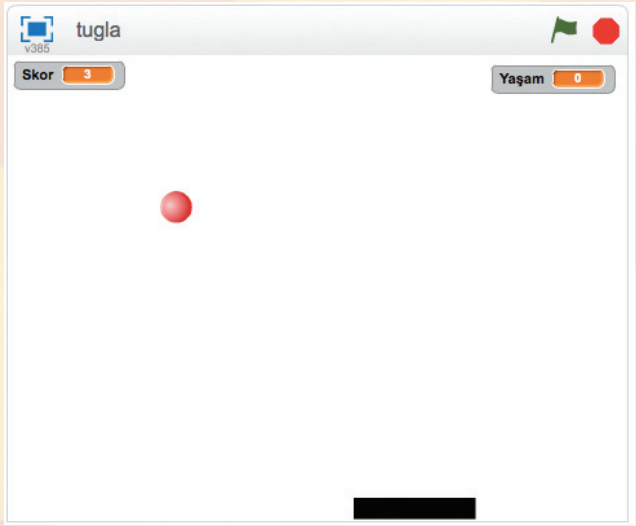
Şekil 3

Şekil 4

Eğer yaşam değişkeni 0 olduysa, oyunu durdurmalıyız. Bunun için, sürekli olarak yaşam değişkeninin sıfıra eşit olup olmadığını kontrol etmeli, eğer 0 ise **tümünü durdur** komutuyla programı durdurmalıyız. (Şekil-5)

Şekil 5

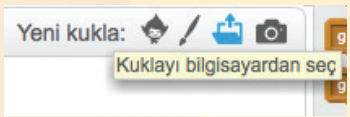
Yaşam değişkenini sürükleyip bırakarak istediğiniz yerde görünmesini sağlayabilirsiniz. (Şekil-6)



Şekil 6

Tuğlalarımızı Oluşturalım

Şimdi, tuğlalara ihtiyacımız var. Birden çok tuğlamız olacağı için bunları tek tek kopyalamak yerine, ikizini oluşturmayı tercih edeceğiz. Bunun için, kontrol sekmesi altındaki ikizini oluştur komutunu kullanacağız. Öncelikle, bir tane tuğla oluşturalım. İster kendi tuğlanızı oluşturun, isterseniz ekte bulunan **tuğla** karakterini programın içine aktarın. (Şekil-7)



Şekil 7

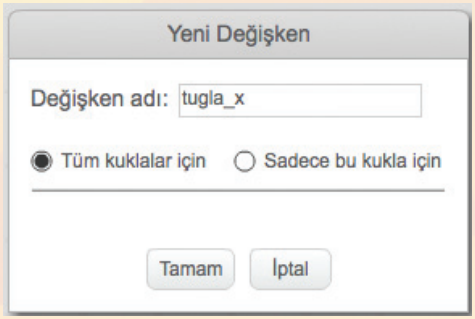
Tuğla karakterine gelip yeşil bayrak tıklanınca, asıl karakterinin görünmemesini fakat ikizlerinin görünmesini sağlayalım. (Şekil-8)

Şekil 8

İlk tuğlayı, olmasını istediğimiz yere getirip, yeşil bayrak tıklanınca, tuğlanın o koordinatlarda olmasını sağlayalım. (Şekil-9)

Şekil 9

Ardından tuğlalarımızın apsisi, yani x koordinatını belirlemek için bir değişken oluşturalım (tuğla_x adını verdik). (Şekil-10)

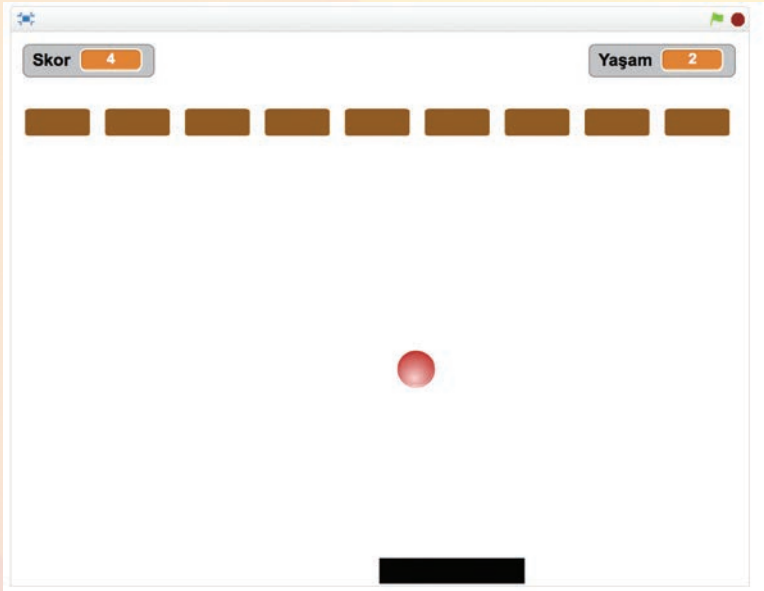


Şekil 10

tuğla_x değişkenimizin ilk değerini atayalım ve her bir ikiz başlatıldığında, tuğla_x değişkenini 52 artırıp tuğlamızın kopyasının o noktaya gitmesini sağlayalım. 9 tane tuğla oluşturmak istediğimiz için, 9 kere dönecek bir döngü içerisinde ikizini oluşturalım. Ve oluşan her ikizin, tuğla_x değişkeninin bulunduğu konuma gitmesini sağlayalım. (Şekil-11)

Şekil 11

Yeşil bayrağa tıklayalım. Tuğlalarımızın, bir sıra dizilmesi gerekiyor.
(Şekil-12)



Şekil 12

Topumuz tuğlalara değdiğinde, tuğlamızın yok olmasını istiyoruz. Bunu, tuğla ikizlerini oluşturduğumuz kod bloğunda yapabiliriz. Sürekli olarak, topun tuğlaya değip değmediğini kontrol edelim. Tuğla topa değdiği an, tuğlanın ikizini silelim. (Şekil-13)

Şekil 13

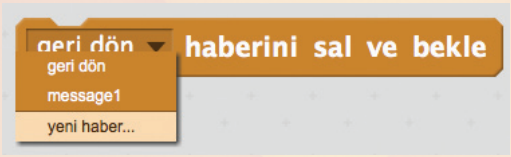
Haber Gönderelim

Şu an oyunumuzu çalıştırdığınızda, tuğlaya değen top geri sekmeden tuğlayı yok edip devam ediyor. Topun, tuğlaya değdiğinde geri sekmesini sağlayalım. Bunun için, top tuğlaya değdiğinde topumuza **geri dön** diye bir haber göndereceğiz. Bunun için, **olaylar** sekmesinin altında bulunan **... haberini sal ve bekle** komutunu kullanacağız.

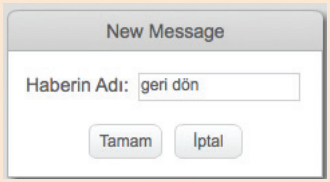
(Şekil-14)

Şekil 14

Geri dön adlı yeni bir haber oluşturacağız. (Şekil-15,16)



Şekil 15



Şekil 16

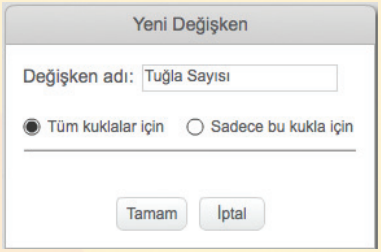
Top tuğlalardan birine değdiğinde, **geri dön** haberini gönderecek. (Şekil-17)

Şekil 17

Top karakterine gelip geri dön haberini alan topun **180-yön** yönüne dönmesini sağlayalım. (Şekil-18)

Şekil 18

Peki, tuğlaların bittiğini nasıl anlayacağız? Öncelikle, **tuğla sayısı** adlı bir değişken oluşturalım. (Şekil-19)



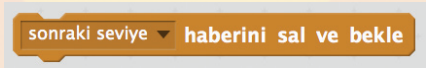
Şekil 19

Başlangıç değerini 0 olarak atayıp her tuğla ürettiğimizde, **tuğla_sayısı** değişkenini 1 artıralım. (Şekil-20)

Şekil 20

Sonraki seviye adlı bir haber oluşturalım. (Şekil-21)

Şekil 21



Top tuğlalara her değdiğinde, tuğla sayısını bir azaltalım ve bu sayı sıfır olduğunda, oyunun bir sonraki seviyeye geçmesini sağlayalım. (Şekil-22)

Şekil 22

Şimdilik, sadece iki seviye yapacağız. Sonraki seviye haberi alındığında, tuğlalarımızın iki sıra olmasını sağlayacağız. Siz dilerseniz, farklı seviyeler de tasarlayabilirsiniz. 9 tuğlayı bir önceki seviyedeki gibi oluşturduktan sonra, diğer tuğlaların y koordinatını 30 aşağıya alıp x koordinatını yine -210 değerini atıyoruz. (Şekil-23)

Şekil 23

Geri Sayım

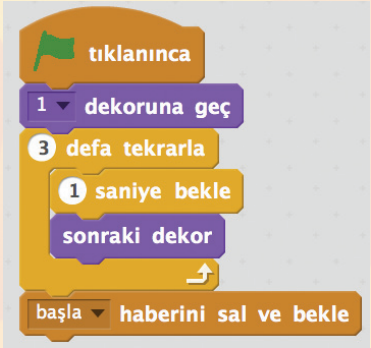
Şimdi de, oyunumuzun başlangıcına bir giriş ekranı ekleyelim. Bunu, birçok farklı şekilde gerçekleştirebiliriz. Biz yeşil bayrağa tıkladığımızda, 3, 2, 1 diyerek gelen bir ekranın ardından oyunumuzu başlatacağız. Öncelikle, sahne üzerine gelelim ve yeni bir sahne oluşturalım. Yeni dekorumuzun adını 3 yapalım ve ekranı kaplayacak şekilde bir 3 sayısı yazalım. (Şekil-24)

Şekil 24

Daha sonra, bu dekordan 2 tane daha kopya alıp 3 sayısını 2 ve 1 diye değiştirip dekor isimlerini ise 3 sayısı, 2 sayısı, 1 sayısı ve arka plan olacak şekilde isimlendirelim.
(Şekil-25)

Şekil 25

Yeşil bayrağa tıkladığımızda, öncelikle 3, 2 ve 1 sayılarının görünmesini ve ardından boş sahne geldiğinde oyunun başlamasını istiyoruz. O halde, sahne kısmında yeşil bayrak tıklandığında, 3 dekoru ile başlayıp sırasıyla diğer sahneleri 1 saniye aralıkla gösterelim ve ardından **başla** diye bir haber gönderelim. (Şekil-26)



Şekil 26

Başla haberi gelene kadar, diğer karakterlerin görünmesini istemiyoruz. Diğer karakterlerde yeşil bayrak olan yerleri, **başla haberi gelince** olayı ile yer değiştirelim. (Şekil-27, 28 ve 29)

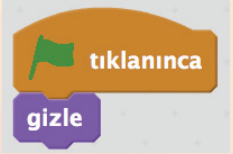
Şekil 27

Şekil 28

Şekil 29

Top ve raket karakteri yeşil bayrak tıklanınca görünmemeli, başla haberi ile görünmeye başlamalıdır. Raket karakteri için yapılması gereken değişiklik: (Şekil-30, 31)

Şekil 30



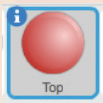
Şekil 31

Top karakteri için yapılması gereken değişiklik: (Şekil-32, 33)

Şekil 32

Şekil 33

BU BÖLÜMDE KULLANILAN KOD BLOKLARI



Tuğla

sonraki seviye haberi gelince
Tuğla Sayısı , 0 olsun
2 defa tekrarla
tugla_x , -210 olsun
9 defa tekrarla
Tuğla Sayısı 'i 1 arttır
kendim in ikizini yarat
y'yi -30 arttır

tıklanınca
x: -210 y: 120 noktasına git
büyüklüğü % 50 yap
gizlen
Tuğla Sayısı , 0 olsun
tugla_x , -210 olsun
9 defa tekrarla
Tuğla Sayısı 'i 1 arttır
kendim in ikizini yarat

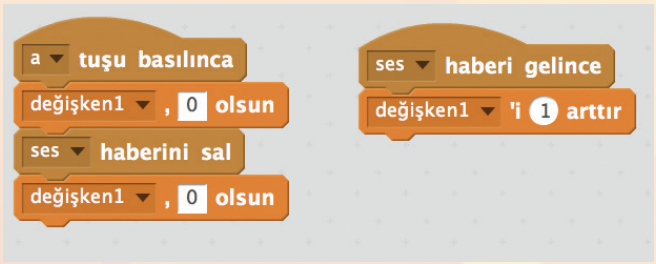
ikiz olarak başladığımda
görün
x, tugla_x olsun
tugla_x 'i 52 arttır
sürekli tekrarla
eğer Top a değdi (mi?) ise
geri dön haberini sal ve bekle
bu ikizi sil
eğer Tuğla Sayısı = 0 ise
sonraki seviye haberini sal ve bekle

ÇALIŞMA SAYFASI

1. Aşağıdaki kod bloklarını çalıştırıp Yeşil Bayrak altında ikiz oluşturmakla, bir olay komutu altında ikiz oluşturmak arasındaki farkı belirtiniz.

2. Aşağıdaki kod blokları arasındaki farkı belirtiniz.

3. Aşağıdaki kod bloklarına göre, **A** tuşuna bastıktan sonra **Değişken1** adlı değişkenin son değeri ne olur?



Akvaryum

Bu Aktivitede Neler Öğreneceğiz?

- İkiz oluşturmayı,
- Rastgele sayıları kullanmayı,
- Bir karakterin fare okunu takip etmesini,
- Değişkenlerin farklı görünümlerini kullanmayı.

Uygulamaya başlamadan önce, kedimizi silmeyi unutmayalım.

BİLGİ

Scratch ile, dışarıdan aldığınız bir ses üzerinde düzenleme yapabilirsiniz.

EK GÖREV

- Puan arttıkça balıklar hızlansın.
- Bazı balıklar daha büyük, bazı balıklar daha küçük olsun.

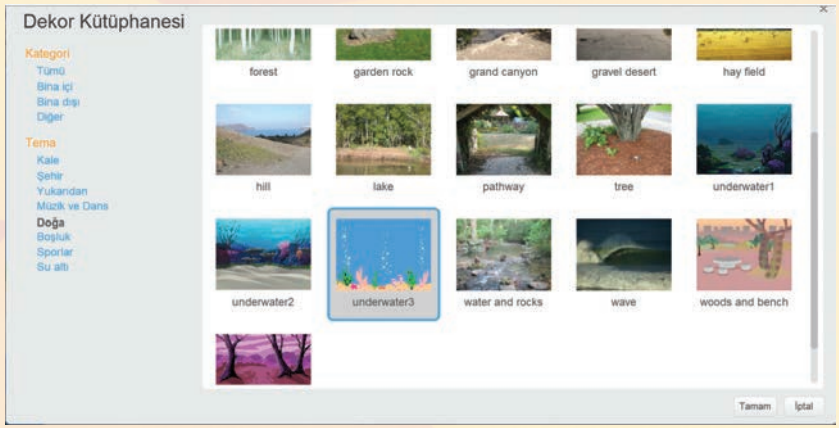
Akvaryum Seçelim

Bu bölümde, bir akvaryumumuz ve içinde bir sürü balığımız olsun istiyoruz. Balıklarımızı tek tek kopyalamak ve kodlamak, oldukça uzun ve zahmetli bir iş. Bunun yerine, Scratch 2.0 ile birlikte gelen Klonlama (ya da iki-zini oluşturma) özelliğini kullanacağız.

Öncelikle, akvaryum için bir sahne seçelim. Dekor sekmesinin altındaki "Dekoru kütüphaneden seç" düğmesine tıklayalım. (Şekil-1)

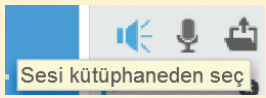
Şekil 1

Temalardan **Doğa**'yı seçip "underwater" serisinden birini arka planımız olarak belirleyelim. Biz underwater3'ü seçtik. (Şekil-2)



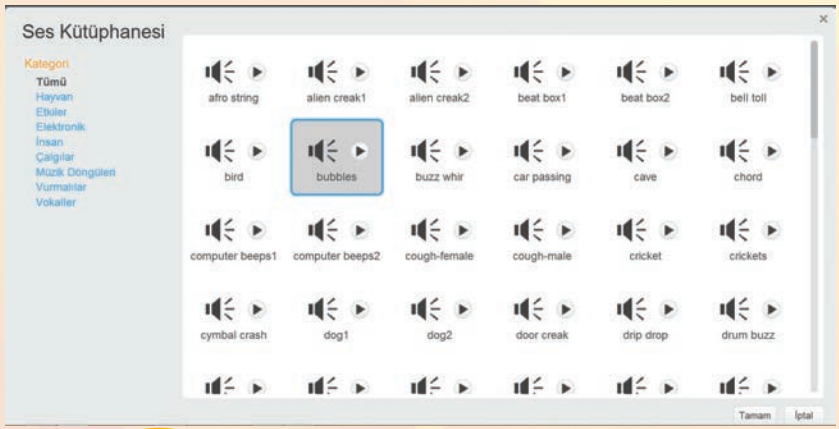
Şekil 2

Akvaryum sahnemiz hazır. Şimdi akvaryumumuza ses ekleyelim. Öncelikle, sahnenin üzerine tıklayalım. Sesler sekmesinden, sesi kütüphaneden seç düğmesine tıklayalım. (Şekil-3)



Şekil 3

Ardından, ses kütüphanesinden bubbles'ı seçelim. (Şekil-4)



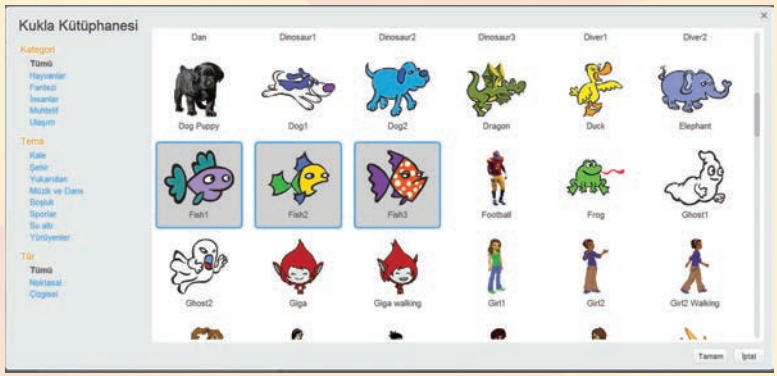
Şekil 4

Baloncuk sesinin sürekli tekrar etmesini istiyoruz. Yeşil bayrak tıklanınca, sürekli tekrarla içinde "bubbles" sesini çalalım. (Şekil-5)

Şekil 5

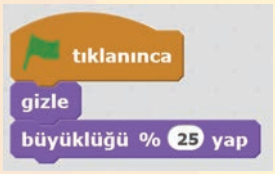
Akvaryuma Balıkları Atalım

Yeşil bayrağa tıkladığınızda, baloncuk sesleri gelmeli. Akvaryumumuzun içi şu anda boş, buraya birkaç balık ekleyelim. Ya internetten bulduğunuz balık resimlerini ekleyebilir (internetten bulunan dosyalar gif veya png olup etrafı transparan olmalı) ya da Scratch'in kendi kütüphanesindeki balıkları içeri aktarabiliriz. Scratch kütüphanesinden 3 tane balığı sahneye ekleyelim. Kuklayı "**kütüphaneden seç**" düğmesine tıklayıp Fish1, Fish2 ve Fish3 balıklarını seçelim. (Şekil-6)



Şekil 6

Bu balıkları kopyalayarak çoğaltabiliriz fakat 30 tane balık oluşturmak için, her birini kopyalamalıyız ve her biri için yapmak istediğimiz bir değişiklik olduğunda, ayrı ayrı kodlarla uğraşmalıyız. Bunun yerine, Scratch 2.0 ile gelen yeni bir özelliği, ikizini oluşturma'yı (klonlama) kullanabiliriz. Fish1'in kod sahnesine gelelim. Öncelikle, yeşil bayrak tıklanınca balığımız görünmez olsun. Ve, balığımızın orijinal hali çok büyük olduğundan ¼'ü yani %25'i olacak şekilde küçültelim. (Şekil-7)

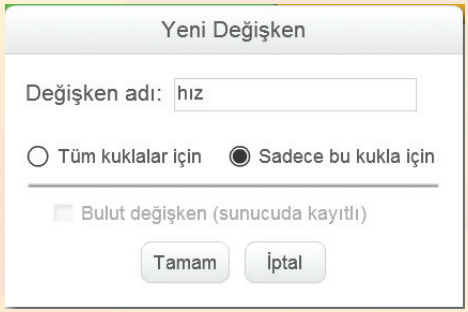


Şekil 7

Aynı balıktan, istediğimiz kadar ikiz üretebiliriz. Biz şimdilik, bir balıktan 10 tane ikiz oluşturacağız. Bu on ikizi aynı anda oluşturmayıp araya rastgele bir süre ekliyoruz ki hareketler tamamen aynı olmasın. Yeşil bayrak tıklandığında, 10 kere dönecek bir döngü ile ikizler oluşturuyor ve her ikiz oluştuktan sonra, .1 ile 1 saniye arasında rastgele bir süre bekleyelim. (Şekil-8)

Şekil 8

İkizimizi ürettiğimizde, başlayacağı x değeri (apsisi) ekranın en solu olan -240 olacak, y değeri (ordinatı) ise rastgele bir değer olarak atanacak. Aynı zamanda, tüm balıklarımızın hızının aynı olmasını istemiyoruz. O halde, öncelikle hız adlı bir değişken oluşturalım. Fakat, burada dikkat etmemiz gereken bir şey var: Hız değişkeni, tüm kuklalar için değil, bir kuklaya özel olarak oluşturulmalı. (Şekil-9)



Şekil 9

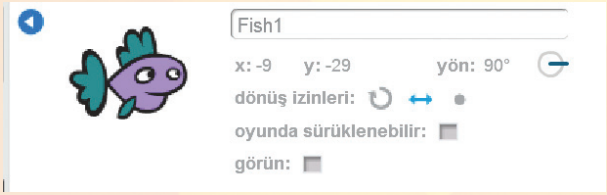
Böylece, hız adlı değişkeni, oluşturulan her bir ikiz için tanımlamış oluyoruz. Her bir balığın ikizini oluşturduğumuzda, önce hız değişkenine 0.5 ile 1.5 arasında rastgele bir sayı atayalım. Daha sonra, şu an için görünmez olan balık karakterini görünür hale getirelim. Koordinatı, balığımız konum olarak ekranın en solunda (yani apsisi -240) ama y ekseninde rastgele herhangi bir yerde olacak şekilde ayarlayalım. Ardından, **sürekli tekrarla** ile hız kadar ilerlemesini ve kenardaysa geri gelmesini sağlayalım. (Şekil-10)

Şekil 10

Programı çalıştırdığınızda, balıkların geri dönerken ters döndüğünü göreceksiniz. Bunu düzeltmek için, Fish1 karakterimizin sol üstünde bulunan "i" harfine tıklayalım.
(Şekil-11)

Şekil 11

Açılan ekranda menüden dönüş izinleri kısmında ortada bulunan sadece "sağa sola" ikonunu tıklayalım.
(Şekil-12)



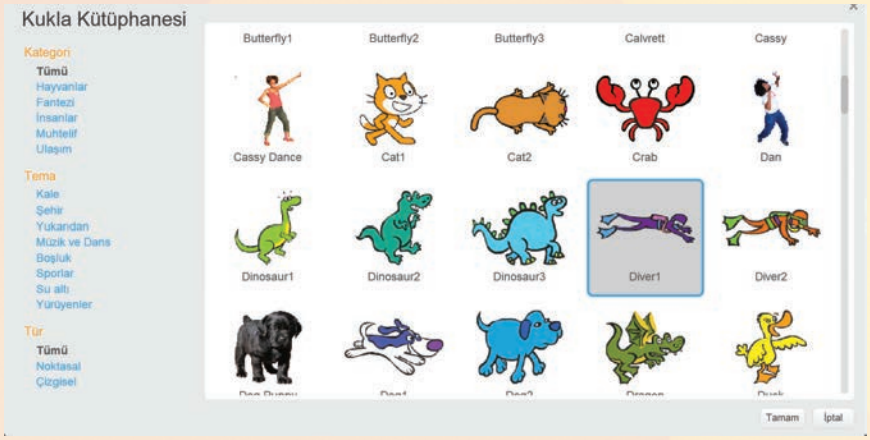
Şekil 12

Böylece, artık balıklarımız ters dönmeyecekler. Aynı şeyleri diğer balıklarımıza da uygulayıp kodlarımızı kopyalayalım. Artık akvaryumumuz, seyretmek için hazır. (Şekil-13)

Şekil 13

Dalgıç Akvaryuma Giriyor

Şimdi, seyirlik bir akvaryum değil, interaktif bir akvaryum uygulaması yapalım. Bir dalgıç, balıkları toplasın ve belirli saniye aralıklarla, rastgele balıkların ikizlerini oluşturup yine belirli bir süre içerisinde dalgıcın kaç tane balık topladığına bakalım. Öncelikle, yeni bir Scratch dosyası açıp sahne arka planına underwater3'ü ekleyelim. Ardından, sahnemize bir dalgıç karakteri ekleyelim. (Şekil-14)



Şekil 14

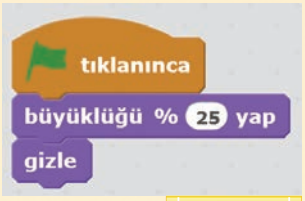
Eklediğimiz dalgıç, ekran boyutları için biraz büyük olduğundan, onu küçültelim. (Şekil-15)

Şekil 15

Dalgıcımızın faremizi süzülerek takip etmesini, fakat fare okumuza değmemesini istiyoruz. Dalgıcımız, sürekli olarak fare okuna doğru dönüp arada bir mesafe bırakacak şekilde fare okunu takip etmeli. Bunun için, aradaki mesafe 50'den küçükse geri gitmesini, 70'den büyükse fare okuna doğru gitmesini istiyoruz. Eğer değer bu iki sayının arasında kalıyorsa, dalgıç olduğu şekilde kalacaktır. (Şekil-16)

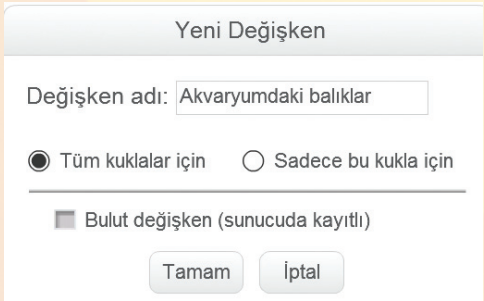
Şekil 16

Dalgıcımız artık faremizi takip ederek yüzebiliyor. Şimdi balıklarımızı ekleyelim. Öncelikle, bir balık üzerinde tüm işlemlerimizi bitirip diğer balıklarımıza kopyalama yapalım. Scratch kütüphanesinden, "fish1"i kendi sahnemize aktaralım. Boyutu büyük olduğu için küçültelim. Diğer balıkları bu balığımızdan klonlayacağımızdan, bu balığın görünmesine gerek yok. (Şekil-17)



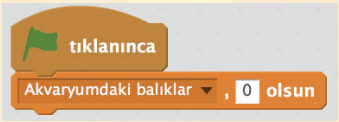
Şekil 17

Belirli aralıklarla balıkların ikizinin oluşmasını istiyoruz. Bu arada, havuzda bulunan toplam balık sayısını da belirlemek istiyoruz. "Akvaryumdaki balıklar" adlı bir değişken oluşturalım. (Şekil-18)



Şekil 18

Başlangıç değerini 0 olarak atayalım. Buna benzer ilk değer atamalarını, istediğimiz her yerde yapabiliriz. Ama biz, daha düzenli olması açısından, bunun gibi programının genelini etkileyen kodlamaları sahnenin üzerinde yapacağız. Sahne alanına gelip "Akvaryumdaki balıklar" değişkenini sıfırlayalım. (Şekil-19)

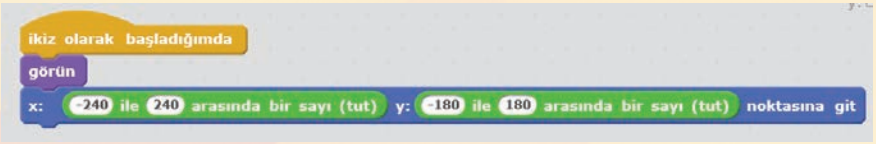


Şekil 19

Ardından, fish1 karakterimize gelip her 2 saniyede bir (süreyi değiştirebilirsiniz) yeni bir ikizini oluşturalım. (Şekil-20)

Şekil 20

"Akvaryumdaki balıklar" değişkenini 1 artıralım. Fish1 karakterimizin ikizi oluştuğunda, oluşan karakter önce ekranda rastgele bir yerde görünmeli. (Şekil-21)



Şekil 21

Ardından, yüzmeye başlamalı ve kenara geldiğinde ters dönmeli. (Şekil-22)

Şekil 22

Balığımızın baş aşağı dönmesini istemiyorsak, daha önce yaptığımız gibi, özelliklerine girip dönüş izinlerini ayarlamalıyız. Dalgıcımız ve balıklarımız yüzüyor, şimdi dalgıcımızla balıkları yakalayalım. Akvaryumdaki oluşturduğumuz balık sayısını tuttuğumuz gibi, dalgıcımızın yakaladığı balık sayısını "puan" olarak tutabiliriz. Puan değişkenini oluşturalım. (Şekil-23)

Şekil 23

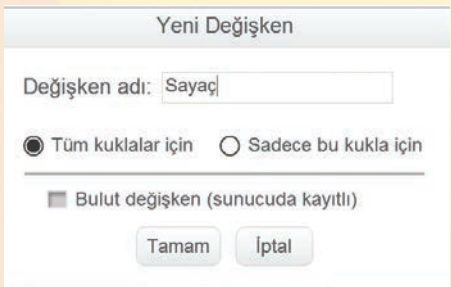
Değişken oluşturur oluşturmaz yapmamız gereken ilk şey, bu değişkeni sıfırlamak. Sahne üzerine gelip kod kısmına ekleme yapalım. (Şekil-24)

Şekil 24

Sürekli olarak, dalgıcın balıklardan herhangi birine değip değmediğini kontrol etmemiz gerekiyor. Eğer dalgıcımız balığa değmişse, akvaryumdaki balık sayısını 1 azaltalım ve puanımızı 1 artıralım. (Şekil-25)

Şekil 25

Fish1 için yazdığımız kodlarımızı, **Fish2** ve **Fish3** içerisine kopyalayalım. Oyunumuzu oynamaya başlayabiliriz. Dilerseniz, oyunumuza biraz daha heyecan katalım ve oyunun bitişi için süre ekleyelim. Örneğin, 1 dakika yani 60 saniye boyunca dalgıç balıkları yakalamaya çalışsın ve süre bitiminde oyun dursun. Bunun için, önce kendi sayacımızı yapalım. **Sayaç** isimli bir değişken oluşturalım. (Şekil-26)



Şekil 26

Değişkenimizin üzerine tıklayıp büyük şekilde görünmesini sağlayalım. (Şekil-27)

Şekil 27

Sayacımızın ilk değerini 60 olarak belirleyip geriye doğru saymasını sağlayalım. Sayaç, 1'den küçük olana kadar (yani 0 olana kadar) sayacımız 1 azalsın ve 1 saniye beklesin. Sayaç 0 olduğunda, hepsini durdur komutuyla oyunu tamamen durduralım. (Şekil-28)

Şekil 28

BU BÖLÜMDE KULLANILAN KOD BLOKLARI

Fish2

tıklanınca
büyüklüğü % 25 yap

ikiz olarak başladığımda
görün
x: -240 ile 240 arasında bir sayı (tut) y: -180 ile 180 arasında bir sayı (tut) noktasına git
sürekli tekrarla
2 adım git
kenara geldiysen sek
eğer diver a değdi (mi?) ise
gizlen
puan 'i 1 arttır
Havuzdaki balıklar 'i -1 arttır

tıklanınca
büyüklüğü % 25 yap
gizlen
sürekli tekrarla
2 saniye bekle
kendim in ikizini yarat
Havuzdaki balıklar 'i 1 arttır

Fish3

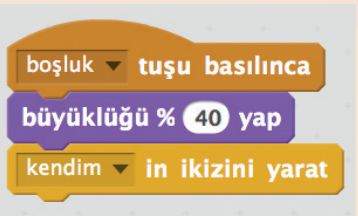
ikiz olarak başladığımda
görün
x: -240 ile 240 arasında bir sayı (tut) y: -180 ile 180 arasında bir sayı (tut) noktasına git
sürekli tekrarla
2 adım git
kenara geldiysen sek
eğer diver a değdi (mi?) ise
gizlen
puan 'i 1 arttır
Havuzdaki balıklar 'i -1 arttır

tıklanınca
büyüklüğü % 25 yap
gizlen
sürekli tekrarla
2 saniye bekle
kendim in ikizini yarat
Havuzdaki balıklar 'i 1 arttır

ÇALIŞMA SAYFASI

1. Aşağıda verilen kod bloklarının sonunda "sayı1" ve "sayı2" değişkenleri eşitlendiğine göre, kod bloklarında "..." ile bırakılan boşluğa gelmesi gereken sayıları bulunuz.

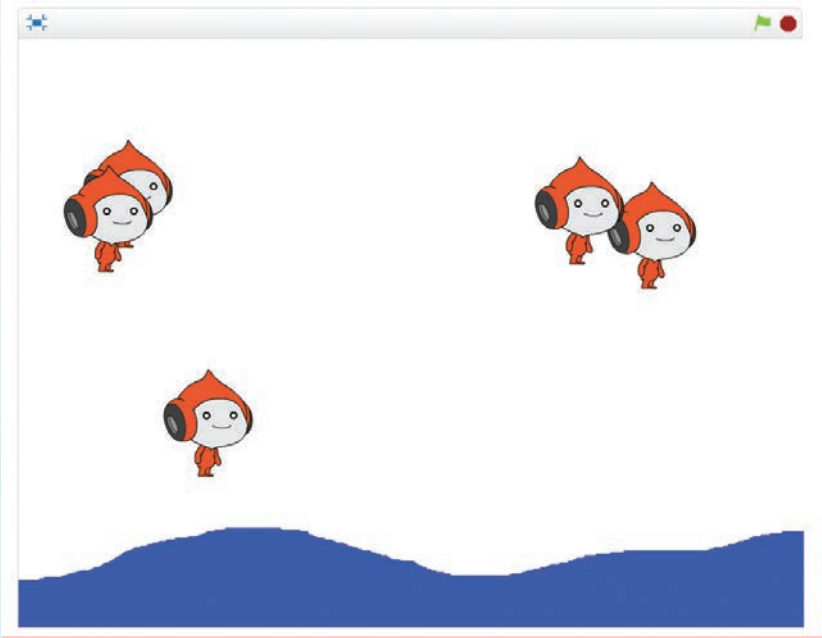
2. Boş bir sahneye balık karakterlerinden birini aktarın. **Hız** adlı bir değişken oluşturun ama değişkeni **sadece bu kukla** için oluşturun. Aşağıdaki kod bloklarını oluşturup boşluk tuşuna bastığımızda nasıl bir olay olmasını bekliyorsunuz? Eğer hız değişkenini **sadece bu kukla** için değil **tüm kuklalar** için oluşturduğumuz, nasıl bir fark olurdu?



Hareket ve Ses Algılama

Bu Aktivitede Neler Öğreneceğiz?

- Bilgisayarımızdaki web kamerasını kullanmayı,
- Kamera ile karakter hareket ettirmeyi,
- Haber alıp göndermeyi,
- Ses algılamayı.



BİLGİ

Scratch üzerindeki video kod bloğu, karakter ve sahne üzerindeki hareketi ayrı ayrı algılayabilmektedir.

EK GÖREV

- Elimizin hareketine göre, Scratch üzerindeki sesi artırıp azaltabileceğiniz bir uygulama yazalım.
- Bu bölümde yaptığınız uygulamaya skor ekleyelim.

Scratch üzerinde bulunan algılama sekmesinde, Scratch ile web kameranız ile algıladığınız bazı işlevleri ve mikrofondan gelen sesin şiddetini ölçebiliyoruz. Dilerseniz, bu komutları yakından tanıyalım.

Scratch Üzerinde Web Kamera ile Kullanılabilecek Komutlar

Web kameradan gelen görüntüleri algılamak için 3 tane kod bloğumuz var. Bunlar, Scratch üzerinde Algılama sekmesinin altında bulunmaktadır. Bu komutları tek tek denemenizi ve ardından uygulamaya geçmenizi tavsiye ediyoruz. (Şekil-1, 2, 3)

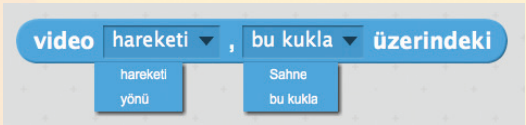
Videoyu aç komutu, web kamerasını açıp kapatmayı ya da sağ ve solun yerini değiştirmesini sağlar.

Şekil 1

Video saydamlığı komutu, web kameradan gelen görüntünün saydamlığını ayarlar. Bu saydamlık, eğer %100 ise sahnede bir şey görünmeyecek, eğer %0 ise ekranda görüntü tam olacaktır.

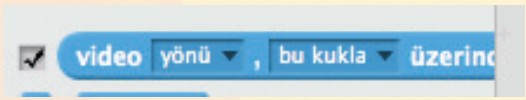
Şekil 2

Yanda görülen kod bloğu ile bir hareketin var olup olmadığını, hatta bu hareketin hızını görebiliriz. Bu hareketin, sahne üzerinde herhangi bir yerde mi, yoksa kod yazdığınız kukla üzerinde mi olduğunu anlayabiliriz. Bu hareketin şiddetinin yanında, yönünü de (sağ veya sol olarak) belirleyebiliriz. Eğer hareketin yönü sağa doğru ise pozitif, sola doğru ise negatif bir değer çıkacaktır.



Şekil 3

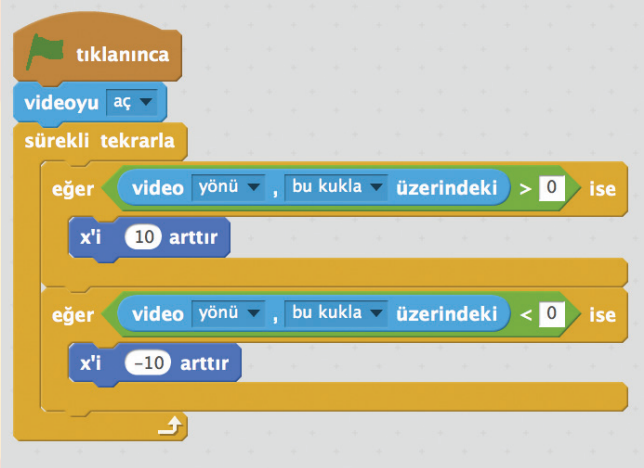
Bu verileri denemek ve verilerin ekranda görünmesi için algılama sekmesinde kod bloğunun yanındaki kutucuğu işaretlemeliyiz. (Şekil-4)



Şekil 4

Kamera ile Kediyi Hareket Ettirelim

Yeşil bayrağa tıklayınca kamerayı açalım. Sonrasında sürekli olarak kameradan gelen hareketin yönünü belirleyelim. Kedinin; hareketin yönü pozitif ise sağa doğru gitmesini, negatif ise sola doğru gitmesini sağlayalım. (Şekil-5)



Şekil 5

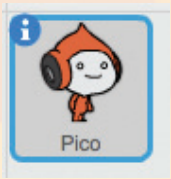
Biraz oynadıktan sonra kedimizin neredeyse hiç duramadığını fark edeceksiniz. Kedinin daha az hareket etmesini sağlamak için hassasiyet ayarı yapmamız gerekecek. Program, yukarıdaki kod bloğuna göre sağa ya da sola doğru en ufak hareketi algılayıp kedimizi sağa ya da sola doğru hareket ettiriyor. Bunu engellemek için **0** olan hassasiyet ayarımızı farklı sayılarla deneyeceğiz. Biz **20** ile güzel bir hareket sağladığımızı düşünüp kod bloğumuzu buna göre değiştirdik. Siz de, farklı sayılarla denemeler yapabilirsiniz. (Şekil-6)

Şekil 6

Pico'lar Düşüyor

Bu uygulamada, ekranın en üst noktasından aşağı düşen Pico karakterleri (dilerseniz başka karakterler de belirleyebilirsiniz) yapacak ve sonra onları kurtaracağız.

Yeni bir proje açıp kedi karakterimizi silelim. Ardından Pico karakterini içeri aktaralım. (Şekil-7)



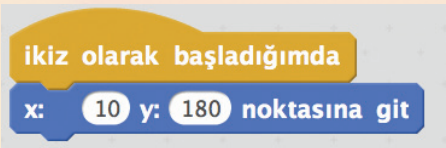
Pico karakterlerinin belirli aralıklarla ortaya çıkmasını ve aşağı düşmesini istiyoruz. Öncelikle, Pico karakterlerini biraz küçültelim ve orijinal Pico'muzu gizleyelim. (Şekil-8)

Şekil 8

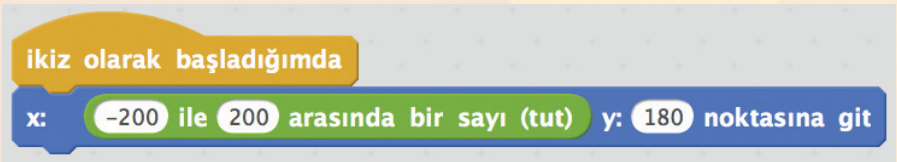
Ardından, rastgele süreler bekleyerek ikiz oluşturalım. Dilerseniz, bu süreyi değiştirerek Pico ikizlerinin daha hızlı veya daha yavaş oluşmasını sağlayabiliriz. (Şekil-9)

Şekil 9

Yeşil bayrağa tıkladığınızda, ekranda hiçbir şeyin değişmediğini göreceksiniz. Oluşturduğumuz ikizlerin görünmesini ve hareket etmesini sağlamalıyız. Öncelikle, oluşan ikizin en tepede rastgele bir yere gitmesini sağlayalım. (Şekil-10, 11)



Şekil 10



Şekil 11

Kopyalarımız artık görünebilir ve en yukarıdan aşağıya doğru yavaşça düşebilir. (Şekil-12, 13)

Şekil 12

Şekil 13

Oluşturduğumuz kod bloğunun tümü şöyle olacaktır: (Şekil-14, 15)

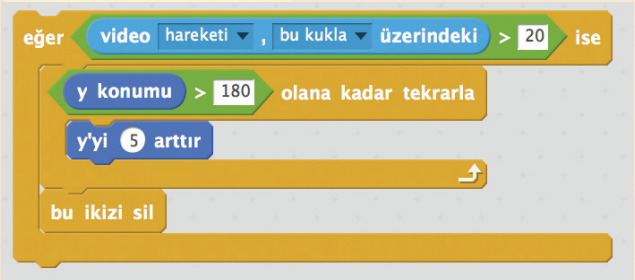
Şekil 14

Şekil 15

Artık Pico'lar yere düşüyor. Pico kuklalarının üzerinde hareket algıladığımızda Pico'ların yukarı kadar geri uçmasını sağlayalım. Bunun için öncelikle, kameramızı açıp kukla üzerindeki hareketi algılayalım. Bundan önce yazdığımız kod bloklarının üzerine eklemeler yapacağız. (Şekil-16)

Şekil 16

Yeşil bayrak açıldığında web kameramız açılacaktır. Fakat henüz Pico'ları kurtarmak için herhangi bir şey yapmadık. Pico'lar aşağıya düşerken onlar üzerinde bir hareket tespit edersek en yukarı doğru çıkmalarını ve en yukarı çıktıklarında yok olmalarını (ikizlerinin silinmesini) istiyoruz. (Şekil-17)



Şekil 17

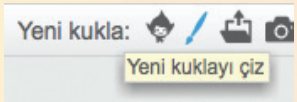
Pico'ları Kurtaralım

Şimdi bu kod bloğumuzu yerine yerleştirelim. Pico'lar aşağıya düşerken üzerinde hareket olup olmadığını kontrol edeceğiz. (Şekil-18)

Şekil 18

Bu uygulamaya puan ve yaşam ekleyebiliriz. Ayrıca Pico'lar boşluğa değil de suya, denize düşüyor gibi çizim yapabiliriz.

Ekranın alt kısmına yapacağımız dalgalar için basit bir çizim yapalım. Bunun için bir karakter oluşturalım. (Şekil-19)



Şekil 19

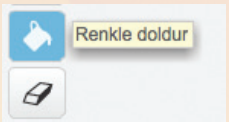
Fırçamızı alıp istediğimiz şekilde bir dalga çizelim. (Şekil-20, 21)

Şekil 20

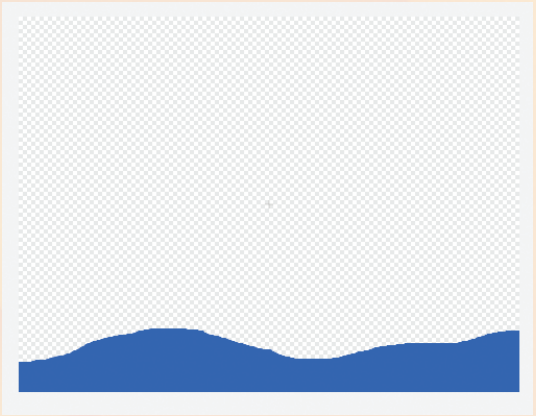
Şekil 21

Eğer renk seçimi yapmadıysanız orta alandan dilediğiniz rengi seçebilirsiniz.

Çizdiğimiz bu dalganın alt tarafını boyayalım. Bunun için İçini Doldur aracını seçip boyamak istediğiniz bölgede herhangi bir noktayı işaretlemeniz yeterlidir. (Şekil-22, 23)

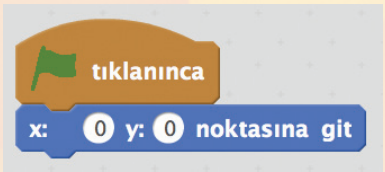


Şekil 22



Şekil 23

Çizdiğimiz bu karakterin kod bloğu oldukça basit olacak. Karakter, uygulamayı geliştirirken el ile sağa sola çekilmiş olabilir. Bunun için oyun başladığında yani yeşil bayrağa tıklandığında karakterimizin ekranı ortalamasını, yani orijinde olmasını istiyoruz. (Şekil-24)



Şekil 24

Dilerseniz, bu oyunumuza ses katarak daha eğlenceli bir hale getirebilirsiniz. Scratch'in kendi kütüphanesinde ses bulabileceğimiz gibi, kişisel kullanımlarınız için **https://freesound.org/** gibi web sitelerinden ses dosyaları indirebilirsiniz.

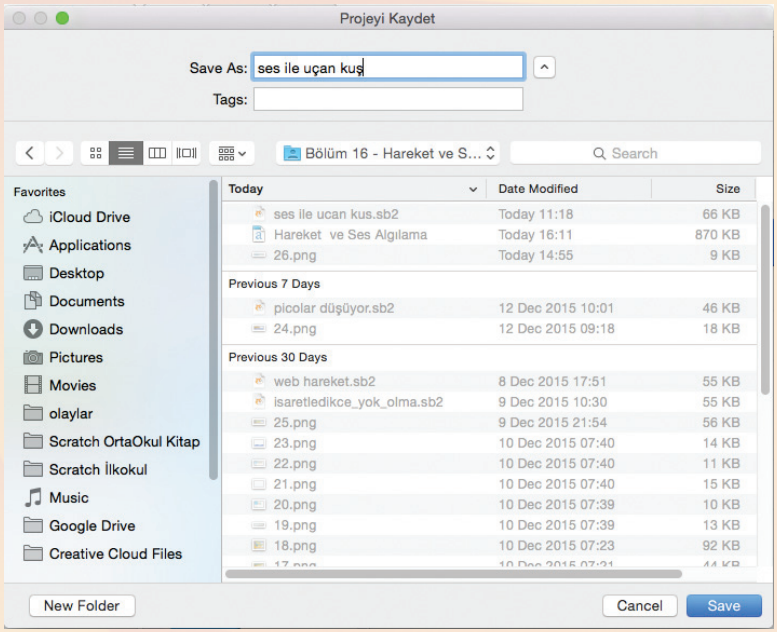
Ses ile Uçan Kuş Oynamak

Flappy Bird adıyla bilinen oyunumuzu Uçan Kuş adıyla daha önceki bölümlerde yapmıştık. Oyun içerisinde, kuşun uçması için boşluk tuşuna basmamız gerekiyordu. Dilerseniz, şimdi yazdığımız uygulamanın üzerinde küçük değişiklikler yaparak ses ile oynanan bir hale getirelim. Öncelikle, algılama sekmemizin altında bulunan **ses şiddeti** kod bloğuna göz atalım. (Şekil-26)

Şekil 26

Ses Şiddeti bloğunun yanında bulunan kutucuğa tıklayarak bu değişkene ait veriyi görebilirsiniz.

Uçan Kuş oyunumuzu açıp **Ses ile Uçan Kuş** adıyla kaydedelim. (Şekil-27)



Şekil 27

Yapacağımız değişiklik **Kuş** karakteriyle ilgili. Bu nedenle, önce Kuş karakterini seçip bu karaktere ait kod blokları üzerinde değişiklik yapacağız. (Şekil-28)

Şekil 28

Öncelikle, boşluk tuşuna bastığımızı algılayan bölüme gelelim. (Şekil-29)

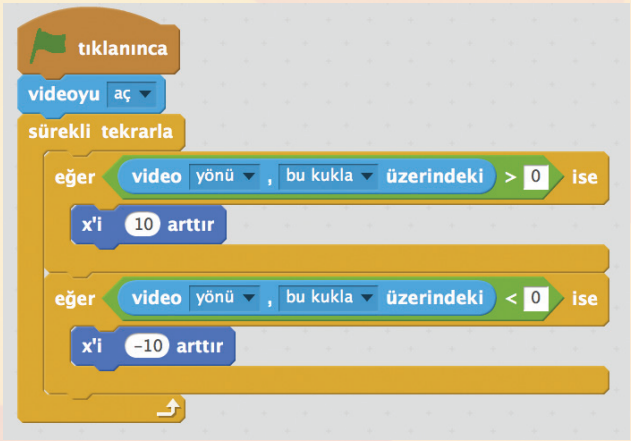
Şekil 29

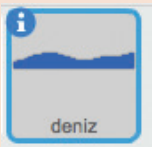
Bu bölümde **boşluk tuşu basılı (mı)?** kısmını değiştireceğiz. Ses şiddeti belirlediğimiz şiddetten fazla ise **uç haberini** göndereceğiz. Bu sayı (ses şiddeti) 0 ile 100 arasında değişen bir sayı olup istediğiniz değeri belirleyebilirsiniz. Bu sayı sizin ses hassasiyetiniz olacaktır. Biz bu ses hassasiyetini 20 olarak belirledik. (Şekil -30)

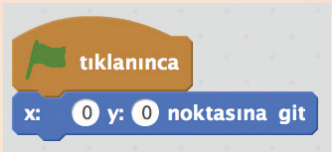
Artık ses ile uçan kuş oyununu oynayabilirsiniz. İyi eğlenceler.

Şekil 30

BU BÖLÜMDE KULLANILAN KOD BLOKLARI

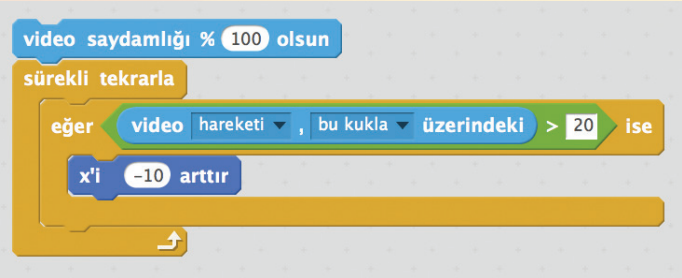






ÇALIŞMA SAYFASI

1. Web kamerasında sağa doğru bir hareket olduğunda karakterin sağa gitmesini isteyen bir öğrenci aşağıdaki kod bloğunu kullanmıştır. Bu kod bloğunda yaptığı hata ya da hataları düzeltiniz.



2. Aşağıda verilen kod bloklarına göre yeşil bayrağa basıldıktan sonra ekranda,

   a. 6 saniye sonra kaç tane karakter görünebilir?

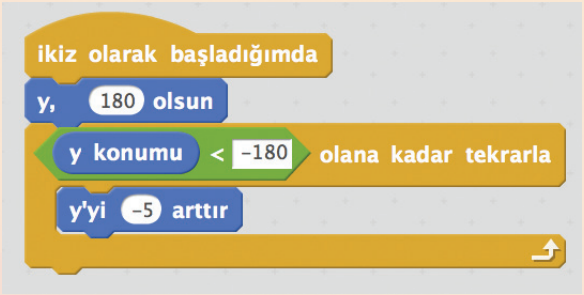
   .....

   b. 18 saniye sonra kaç tane karakter görünebilir?

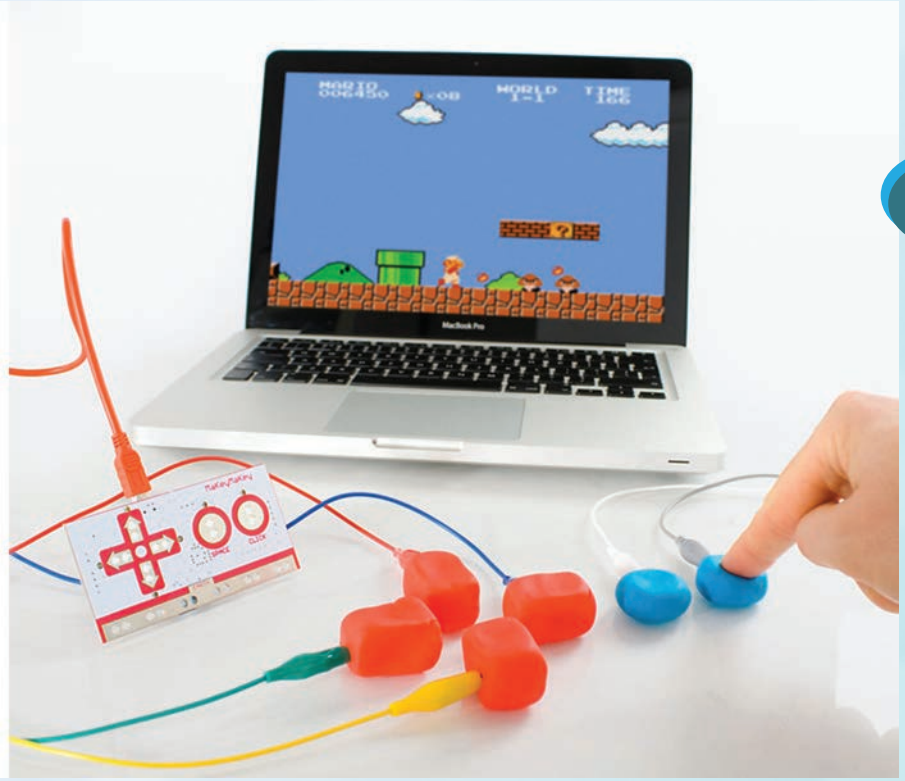
   .....

   c. 30 saniye sonra kaç tane karakter görünebilir?

   .....



Makey Makey



Eğer şu anda internet bağlantınız varsa **www.makeymakey.com** adresinde bulunan videoyu izlemelisiniz. Peki nedir bu MakeyMakey?

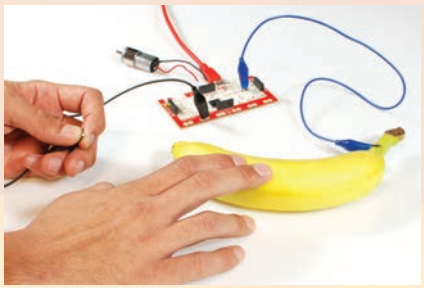
MaKey MaKey ismi Make + key = Makey şeklinde anahtar üret diyebileceğimiz şekilde uydurulmuş bir kelimedir. "Makey Makey 21. yüzyıl için geliştirilmiş bir icat kitidir." diyor Makey Makey'in mucitleri. Makey Makey ile, günlük hayatta kullandığımız hemen hemen her nesneyi dokunmatik yüzeye çevirebiliriz. Bu öyle bir icat kiti ki yeni başlayandan uzmana, mühendisten sanatçıya kadar herkes farklı şekillerde kullanabilir.

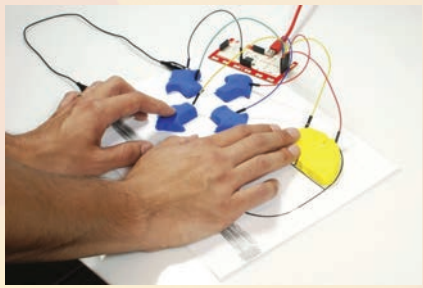
Makey Makey ile Neler Yapabilirim?

Bu size kalmış! Bilgisayarınıza herhangi bir program yükleyin veya herhangi bir web sayfasından bir uygulama açın. Diyelim ki bir müzik programı yüklediniz. Klavye tuşları yerine dilerseniz muz ile piyano çalabilirsiniz ya da siz bunu havuç, patlıcan gibi diğer sebzelerle deneyebilirsiniz.

Pacman oynamak istiyorsunuz ama klavye tuşlarından bıktınız mı? Oyununuzu bunun yerine kurşun kalemlerle oynayabilirsiniz.

Ya da oyun hamurundan kendinize yeni bir joypad oluşturun. Sizin hayal gücü size kalmış.





Nasıl Çalışıyor?

Timsah kabloları kullanarak iki nesneyi birbirine bağlıyorsunuz. Örneğin; siz ve bir muz...

Muza değdiğinizde bir bağlantı kurmuş oldunuz ve Makey Makey bilgisayara bir klavye sinyali gönderdi. Bilgisayar Makey Makey'i bir klavye ve fare olarak görüyor. Bu sayede her program ya da web sitesinde Makey Makey'i rahatlıkla kullanabilirsiniz.

Makey Makey Kimin için?

Sanatçılar, çocuklar, mühendisler, tasarımılar, mucitler, öğretmenler, öğrenciler; kısacası 7'den 70'e herkes için harika bir araçtır.

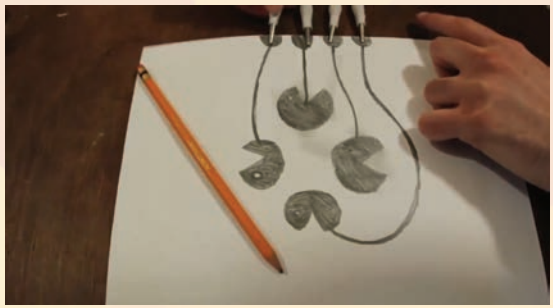
Makey Makey ile ne icat edeceğiniz gerçekten size kalmış.

Makey Makey Hangi Materyallerle Çalışır?

Birazcık bile olsa elektrik ileten her materyal ile çalışır. Kurşun kalem, ketçap, limon, oyun hamuru, sebzeler, parmak boyası vb.

Materyal olarak arkadaşlarınız, kediniz, köpeğiniz, alüminyum folyoya takılmış herhangi bir şey, ıslak herhangi bir şey ve daha birçok şeyi kullanabilirsiniz (kapasitif enerji kullanır, yani sizi elektrik çarpmaz).

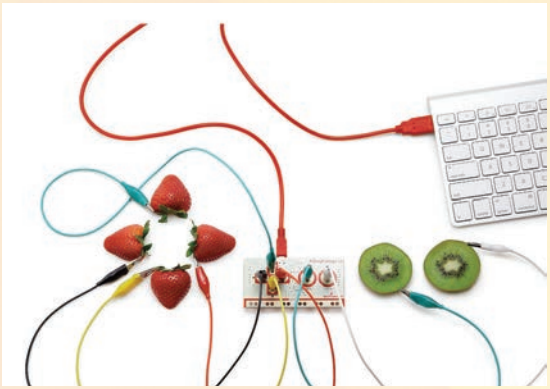
Neden Makey Makey Gibi Bir Ürün Üretildi?



Makey Makey'in yapımcıları herkesin yaratıcı ve mucit olabileceğine inanıyor. Herkesin dünyayı değiştirebilecek potansiyeli olduğunu savunuyorlar. Bundan dolayı, yaşamlarını diğer insanların yaratıcılıklarını daha kolay göstermelerini sağlayacak aletler tasarlamaya adamışlar ve tüm dünyayı bir icat kutusuna benzetiyorlar.

Makey Makey ile yapılmış çok sayıda proje bulabilirsiniz. **www.ogreniyorum.xyz** sitesinde de bu projelerden bazılarını ve nasıl yapıldığını zaman içerisinde takip edebilirsiniz.

Makey Makey'im Var Bu Kitaptaki Projelerle Çalışır mı?



Tabii ki çalışır. Makey Makey hazır bir kittir. Bilgisayarınıza USB kablosu ile Makey Makey devrenizi taktığınız andan itibaren artık sağ, sol, yukarı ve aşağı yön tuşları ve yine Makey Makey üzerinde tüm tuşlar aktif hale gelirler. Tek yapmanız gereken uygun bir düzenek kurup devreyi tamamlamak.

İşte size bir kaç fikir: Oyun hamurlarından bir oyun kolu yapın. Yön tuşlarını bir ucu MakeyMakey üzerinde diğer ucu ilgili oyun kulunun içinde olacak şekilde bir düzenek hazırlayın. Makey Makey toprak (earth) kısmını vücudunuzda herhangi bir yere değdirin. Keyifli oyunlar.

Bir sonraki uygulama için etrafınızda birden çok insan olmalı. Scratch ile yön tuşları, w, a, s, d, f ve g tuşlarını kullanan bir müzik programı yazın. Bir ucu bu tuşlarda diğer ucu insanların elinde olacak şekilde timsah kabloları bağlayın. Ardından, toprak ucu elinize alıp insanlara dokunarak bir orkestra oluşturabilirsiniz.

Bir sonraki kitapta
görüşmek üzere..
Kodlamaya devam!

Çocuklar için SCRATCH ile kodlama

BU KİTAP KİMLER İÇİN?

- DAHA ÖNCE KODLAMA BİLEN YA DA BİLMEYENLER
- ÇOCUĞUYLA BİRLİKTE KODLAMA ÖĞRENMEK İSTEYEN EBEVEYNLER
- KODLAMA ÖĞRETMEK İSTEYEN ÖĞRETMENLER
- YARATICI VE SİSTEMATİK DÜŞÜNMEYİ ÖĞRENMEK İSTEYEN HERKES
- KODLAMA ÖĞRENMEK İSTEYEN TÜM ÇOCUKLAR VE GENÇLER
- MERAK DUYGUSUNU DİJİTAL ARAÇLARLA TETİKLEMEK İSTEYEN HERKES

Tüketmekten Üretmeye MAKER Eğitim Hareketi Sürdürülebilir Bir Model Önerisi

21. yüzyılda her meslek grubu için verimlilik ve rekabet adına artık dijital teknolojilerin kullanımı kaçınılmaz. Bunun temelini oluşturan programlama ise süratle ulusal eğitimin bir parçası olmalı. Bununla, okullarda göstermelik ve ne olduğu anlaşılmayan bir ders olarak değil, her detayı planlanmış, öncelikle bu eğitimi verebilecek öğretmenlerin bunu içselleştirmesiyle başlayacak ve yeni kuşağa ders izlemeyi bir külfet değil keyif haline getirecek uzun soluklu bir dönüşümden söz ediyoruz.

15 yıl sonra mevcut mesleklerin yarısı yok olacak ve o dönemde popüler olacak mesleklerin de yarısını bugünden öngörmek zor. Ancak önümüzdeki döneme ilişkin herkesin üzerinde mutabık kaldığı yegane husus, dijitalleşme ve kodlama becerisinin tüm meslek kollarında rekabet ve verimliliği artıracağı yönünde.

İşte şu anda elinizde tuttuğunuz bu kitabın başlıca hedefi ise, (gelecekte hangi mesleği seçerlerse seçsinler) yeni kuşak bireyler için artık kaçınılmaz hale gelen programlamanın temelini atmak ve bu temeli alan öğrencilerin ilk aşamada programlayıcı (coder) ama daha sonra uygulayıcı (maker) ve en sonunda da sorgulayıcı/farklılaştırıcı (hacker) aşamalarına gelmelerini sağlamak.

İsmail Hakkı Polat / Kadir Has Üniversitesi Yeni Medya Bölümü Öğretim Görevlisi

Kodlamanın müfredata girmesiyle birlikte pek çok eğitim kurumunda ders kitabı olarak okutulan kitabımızın veliler ve öğretmenler tarafından daha verimli kullanılması için hazırlanan Öğretmen Kitabı'na ***www.abakuskitap.com*** dosya merkezinden ulaşabilirsiniz.

editor@abakuskitap.com
facebook.com/makeregitimkitapligi
facebook.com/abakuskitap
twitter.com/abakuskitap

abaküs
Türkiye'nin Bilişim Kaynağı
www.abakuskitap.com

ISBN 978-605-9129-21-3
9 786059 129213